

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previa a la obtención del título de: INGENIERO DE
SISTEMAS**

TEMA:
**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN
PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA**

AUTORES:

LIVARDI PAUL SALGADO FLORES
EDGAR DANIEL SIERRA RIVADENEIRA

DIRECTOR:
DANIEL GIOVANNY DÍAZ ORTIZ

Quito, septiembre del 2014

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE TITULACIÓN

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, septiembre del 2014

Livardi Paul Salgado Flores

CC: 1716690969

Edgar Daniel Sierra Rivadeneira

CC: 1720989654

DEDICATORIA

A Dios, por permitirme compartir este logro con mis seres queridos.

A mis padres por el apoyo y amistad incondicional.

A mi abuelita, Maruja Bonilla, quien es mi ejemplo a seguir y la fortaleza que motiva mis ganas de mejorar cada día.

A mis amigos y familiares quienes participaron durante toda la etapa académica.

Edgar Daniel Sierra Rivadeneira

A Dios, por darme la oportunidad de cumplir una de las metas más importantes en mi vida, por darme sabiduría para vencer los obstáculos.

Mi madre Rocío Flores, por sus cuidados, amor, y apoyo para seguir por el camino correcto. Esas enseñanzas contribuyeron a ser lo que soy ahora.

Mi padre Gavino Salgado, que aunque no te encuentres con nosotros, siempre estuviste cuidando de mí. Tus valores y enseñanzas han hecho que sea un hombre de bien. En donde quiera que estés, te estoy eternamente agradecido.

Livardi Paul Salgado Flores

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana por haber constituido parte fundamental en nuestra formación profesional, en especial al Ing. Daniel Díaz, como tutor del trabajo de titulación quien ha orientado, apoyado y corregido el trabajo con un interés y entrega que ha sobrepasado las expectativas esperadas

Edgar Daniel Sierra Rivadeneira & Livardi Paul Salgado Flores

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
PLANTEAMIENTO DEL PROBLEMA	2
1.1 Antecedentes	2
1.2 Descripción del problema.....	3
1.3 Objetivos	4
1.3.1 Objetivo general.	4
1.3.2 Objetivos específicos.	4
1.4 Justificación.....	5
1.5 Alcance	6
1.6 Metodología.....	7
CAPÍTULO 2	10
MARCO TEÓRICO.....	10
2.1 Proceso de recaudación	10
2.1.1 Descripción del proceso.	10
2.2 Herramientas y tecnología.....	11
2.2.1 Gestores de programación.....	11
2.2.2 Gestores de base de datos.....	12
2.2.3 Lenguaje de programación.....	13
2.3 Generalidades	13
2.3.1 Metodología.	13
2.4 Diagramas UML.....	16
2.4.1 Diagramas caso de uso.	17
CAPÍTULO 3	18
ANÁLISIS Y DISEÑO	18
3.1 Recolección de requisitos y necesidades.....	18
3.2 Proceso de registro	19
3.3 Módulos del sistema de facturación	20
3.3.1 Módulo de administrador.	20

3.3.2	Módulo de registro.	21
3.3.3	Módulo de cobro y facturación.	23
3.3.4	Módulo de reportes.	25
3.4	Diagramas.....	25
3.4.1	Modulo Autenticacion.....	26
3.4.1.1	<i>Proceso autenticación.</i>	26
3.4.2	Modelo usuario.	28
3.4.2.1	<i>Proceso creación de usuario.</i>	28
3.4.2.2	<i>Proceso actualización de usuario.</i>	30
3.4.2.3	<i>Proceso eliminación de usuario.</i>	30
3.4.3	Modelo administrador.	32
3.4.3.1	<i>Proceso creación.</i>	32
3.4.3.2	<i>Proceso actualización de administrador.</i>	34
3.4.3.3	<i>Proceso eliminación de administrador.</i>	35
3.4.4	Modelo clientes.	37
3.4.4.1	<i>Proceso creación.</i>	37
3.4.4.2	<i>Proceso actualización.</i>	39
3.4.4.3	<i>Proceso de eliminación de clientes.</i>	40
3.4.5	Modelo perfiles.	42
3.4.6	Modelo medidor.	43
3.4.6.1	<i>Proceso creación.</i>	43
3.4.6.2	<i>Proceso actualización.</i>	45
3.4.6.3	<i>Proceso eliminación.</i>	46
3.4.7	Modelo lectura de información.	48
3.4.7.1	<i>Proceso registro.</i>	48
3.4.8	Modelo eventos.	50
3.4.8.1	<i>Proceso creación.</i>	50
3.4.9	Modelo multas.....	51
3.4.9.1	<i>Proceso creación.</i>	51
3.4.10	Modelo reportes.	53
3.4.10.1	<i>Proceso generación.</i>	53

3.5	Diagrama de base de datos	54
3.5.1	Diagrama entidad relación.	57
CAPÍTULO 4		58
IMPLEMENTACIÓN.....		58
4.1	Codificación	58
4.1.1	Clase: conexión.	58
4.1.2	Clase: parámetros.	60
4.1.3	Clase administradores.	61
4.1.4	Clase: facturas.	61
4.1.5	Clase: medidores.	63
4.1.6	Clase: lecturas.	63
4.1.7	Clase: eventos.....	64
4.1.8	Clase: asistencia.	64
4.1.9	Clase: perfiles.....	65
CAPÍTULO 5		66
PRUEBAS.....		66
5.1	Pruebas funcionales	66
5.1.1	Creación y asignación.	66
5.1.2	Actualización.....	70
5.1.3	Eliminación de registros.....	72
5.2	Pruebas de conexión	73
5.3	Pruebas carga y estrés.....	76
5.4	Test de configuración	76
CONCLUSIONES		84
RECOMENDACIONES		85
LISTA DE REFERENCIAS		86

ÍNDICE DE FIGURAS

Figura 1. Ciclo de vida del RUP	14
Figura 2. Roles de la metodología RUP	16
Figura 3. Módulos del sistema de facturación	20
Figura 4. Modelo de Caso de uso #1, Proceso Autenticación	26
Figura 5. Modelo de Caso de uso, Proceso Creación de Usuario.	28
Figura 6. Modelo de Caso de uso, Proceso Actualización de Usuario	30
Figura 7. Modelo de Caso de uso, Proceso Eliminación de Usuario	30
Figura 8. Modelo de Caso de uso, Proceso Creación de Administrador.	32
Figura 9. Modelo de Caso de uso, Proceso Actualización de Administrador	34
Figura 10. Modelo de Caso de uso, Proceso Eliminación de Administrador	35
Figura 11. Modelo de Caso de uso, Proceso Creación de Clientes	37
Figura 12. Modelo de Caso de uso, Proceso Actualización de Clientes	39
Figura 13. Modelo de Caso de uso, Proceso Eliminación de Clientes	40
Figura 14. Modelo de Caso de uso, Proceso Creación de Perfil	42
Figura 15. Modelo de Caso de uso, Proceso Creación de Medidor	43
Figura 16. Modelo de Caso de uso, Proceso Actualización de Medidor	45
Figura 17. Modelo de Caso de uso, Proceso Eliminación de Medidor	46
Figura 18. Modelo de Caso de uso, Proceso Lectura	48
Figura 19. Modelo de Caso de uso, Proceso Creación de Evento	50
Figura 20. Modelo de Caso de uso, Proceso Creación de Multas	51
Figura 21. Modelo de Caso de Uso – Generación de Reportes	53
Figura 22. Diagrama físico de la base de datos relacionada	55
Figura 23. Diagrama Integro de base de datos	56
Figura 24. Diagrama físico de la base de datos	57
Figura 25. Diagrama físico de la base de datos relacionada	57
Figura 26. Código Fuente – Clase Conexión	58
Figura 27. Código Fuente – Clase Conexión JDBC	59
Figura 28. Código Fuente – Clase Parámetros	60
Figura 29. Código Fuente – Clase Parámetros – Funciones Varias	60
Figura 30. Código Fuente – Clase Factura	61
Figura 31. Código Fuente – Clase Factura – Función Métodos varios	62
Figura 32. Código Fuente – Clase Factura – Función Cálculo	62
Figura 33. Código Fuente – Clase Medidor	63
Figura 34. Código Fuente – Clase Lectura	63
Figura 35. Código Fuente – Clase Eventos	64
Figura 36. Código Fuente – Clase Asistencia	64
Figura 37. Código Fuente – Clase Perfiles	65
Figura 38. Descripción de Equipos Clientes	74
Figura 39. Información relacionada de Equipos	74

Figura 40. Características del Equipo	75
Figura 41. Test de configuración de Webserver Stress tool.....	77
Figura 42. Detección de URL	77
Figura 43. Tiempos de detección de protocolos.....	78
Figura 44. Análisis de carga en espectro.....	78
Figura 45. Resultados de Gestión Usuarios	79
Figura 46. Resultado de conexión a URL	79
Figura 47. Detalle de resultado de Equipo 1	80
Figura 48. Detalle de resultado de Equipo 2	81
Figura 49. Detalle de resultado de Equipo 3	81
Figura 50. Detalle de resultado de Equipo 4:.....	82
Figura 51. Detalle de resultado de Equipo 5	83

ÍNDICE DE TABLAS

Tabla 1. Metodologías existentes.....	7
Tabla 2. Nivel de gestión de clientes	21
Tabla 3. Niveles de registro	23
Tabla 4. Niveles de facturación.....	24
Tabla 5. Escenario de Caso de Uso, Proceso Autenticación.....	27
Tabla 6. Escenario de Caso de Uso, Proceso Creación de Usuario	29
Tabla 7. Escenario de Caso de Uso, Proceso Eliminación de Usuario	31
Tabla 8. Escenario de Caso de Uso, Proceso Creación de Administrador.....	33
Tabla 9. Escenario de Caso de Uso, Proceso Actualización de Administrador	34
Tabla 10. Escenario de Caso de Uso, Proceso Creación de Administrador.....	36
Tabla 11. Escenario de Caso de Uso, Proceso Creación de Clientes	38
Tabla 12. Escenario de Caso de Uso, Proceso Actualización de Clientes	39
Tabla 13. Escenario de Caso de Uso, Proceso Creación de Administrador.....	41
Tabla 14. Escenario de Caso de Uso, Proceso Creación de Perfil	42
Tabla 15. Escenario de Caso de Uso, Proceso Creación de medidor.....	44
Tabla 16. Escenario de Caso de Uso, Proceso Creación de Medidor	45
Tabla 17. Escenario de Caso de Uso, Proceso Eliminación de Medidor	47
Tabla 18. Escenario de Caso de Uso, Proceso Lectura	49
Tabla 19. Escenario de Caso de Uso, Proceso Creación de Eventos	50
Tabla 20. Escenario de Caso de Uso, Proceso Creación de Perfil	52
Tabla 21. Escenario de Caso de Uso, Proceso Creación de Perfil	54
Tabla 22. Prueba Funcional – Proceso Autenticación	66
Tabla 23. Prueba Funcional – Proceso Generación de Reportes	67
Tabla 24. Prueba Funcional – Proceso Creación de Usuarios	67
Tabla 25. Prueba Funcional – Proceso Asignación de Perfiles.....	68
Tabla 26. Prueba Funcional – Proceso Creación de Clientes.....	68
Tabla 27. Prueba Funcional – Proceso Medidores.....	69
Tabla 28. Prueba Funcional – Proceso Creación de Eventos.....	69
Tabla 29. Prueba Funcional – Proceso Registro de Multas	70
Tabla 30. Prueba Funcional – Proceso Actualización de Usuarios.....	70
Tabla 31. Prueba Funcional – Proceso Actualización de Clientes.....	71
Tabla 32. Prueba Funcional – Proceso Actualización de medidores	71
Tabla 33. Prueba Funcional – Proceso Eliminación de Usuarios	72
Tabla 34. Prueba Funcional – Proceso Eliminación de clientes	72
Tabla 35. Prueba Funcional – Proceso Eliminación de Medidores	73
Tabla 36. Descripción de Equipos Servidor y Red	73

ÍNDICE DE ANEXO

Anexo 1. Diagramas múltiples	1
Anexo 2. Instalación y configuración de sistemas base	1
Anexo 3. Manual de usuario	1

RESUMEN

El objetivo principal del desarrollo del sistema presentado es el análisis, diseño e implementación de un sistema de facturación para las comunidades del proyecto Pesillo Imbabura, quienes proporcionaron sus requerimientos y se logró unificarlos con el fin de estructurar e implementar el sistema utilizando herramientas de desarrollo de fácil uso hacia el usuario e integrando con una base de datos confiable y eficaz que al terminarlo fue entregado hacia las comunidades del proyecto, realizando las capacitaciones necesarias tanto del uso como de la instalación hacia las comunidades llegando a cumplir el objetivo señalado

Durante el desarrollo del sistema se realizó las observaciones respectivas de acuerdo a las inquietudes tanto de nosotros como desarrolladores y aceptando las peticiones únicas de cada comunidad, que permitió parametrizar las diferencias distinguidas en base a la facturación actual.

De igual forma se realizó las pruebas respectivas del funcionamiento con personal de la comunidad con el fin de percibir la aceptación y las preguntas frecuentes del uso del sistema elaborado.

En base al estudio realizado se determina que el sistema es funcional y es entregado formalmente a las personas jefes de juntas de la comunidad para su respaldo y distribución del mismo según su criterio.

El sistema entregado es elaborado en base a las necesidades reportadas, señalando que las herramientas utilizadas han sido adquiridas de forma legal y de acuerdo a los derechos de autor o en su caso de gratuidad de uso.

Cualquier modificación al sistema base entregado actualmente es responsabilidad de la persona solicitante.

ABSTRACT

The objective of the project is the analysis, design and implementation of a billing system for project communities Pesillo Imbabura, who provided us with your requirements and is able to unify in order to structure and implement the system using development tools easy to use and integrate the user with a base of reliable and efficient data to finish it was delivered to the project communities, performing the necessary training from such use or installation to communities coming to meet the objective

During the development of the system corresponding observations according to the concerns both of us as developers was performed and accepting the demands of each unique community, which allowed us to parameterize the differences based on current billing.

Similarly the respective performance tests with community personnel in order to receive acceptance and answer about their questions.

Based on the study it is determined that the system is functional and is formally handed over to the boss of the community together to support and distribute the same at its discretion.

The delivered system is developed based on the needs reported, noting that the tools used have been acquired legally and according to copyright or possibly of free use.

Any modifications to the base system delivered are currently the responsibility of the applicant.

INTRODUCCIÓN

El análisis, diseño e implementación del proyecto busca mejorar la forma de cobro de las comunidades del proyecto Pesillo Imbabura quienes a través del sistema logran obtener un control y registro de la facturación de sus comunidades.

La distribución de este trabajo se realizó de la siguiente forma:

En el capítulo 1: Aborda temas relacionados con la estructura del proyecto, en las que se identifica el título del proyecto, justificación, alcance, el planteamiento del problema, objetivos planteados, las técnicas y métodos que se utilizaron para el desarrollo del proyecto.

En el capítulo 2: Plantea las herramientas y procesos que se debe seguir para el propósito del proyecto describiendo su metodología y el diagrama base,

En el capítulo 3: Describe los módulos y diagramas generados en la etapa de análisis y diseño para la construcción del proyecto.

En el capítulo 4: Constituye la creación del código fuente de los diferentes módulos planteados.

En el capítulo 5: Presenta las pruebas y resultados en base a las ejecuciones realizadas demostrando así el funcionamiento del proyecto.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

El proyecto de agua potable Pesillo-Imbabura nació en 1996. Fue el Consejo Provincial de Imbabura el que efectuó los estudios técnicos con el aporte económico de organismos internacionales como la Corporación Andina de Fomento (CAF) y Banco Interamericano de Desarrollo (BID). Las lagunas de San Marcos serán el punto de donde se alimente de agua las reservas del proyecto. Previamente, se tendrían que realizar trabajos para recolectar parte de las aguas de los ríos Cosungo, San Jerónimo 1 y 2, Arturo, y Boquerám.

Se consideraba iniciar la construcción del proyecto en el año 2006, para estar en funcionamiento a mediados del 2008, sin embargo no fue hasta enero del 2014, en que se realizó la entrega de los estudios definitivos.

Esta obra que beneficiará a dos provincias es una realidad. Se encuentra al norte de Cayambe, pasando el río La Chimba, en la cima de una colina, con una alta presencia de población indígena.

El proyecto dotará de agua potable a la región norte de Pichincha y sur de Imbabura hasta el año 2030. Contempla la rehabilitación de 52 sistemas de agua potable para más de 150 comunidades. Es multidisciplinario, cuenta con estudios viales, estructurales, electromecánicos, eléctricos, y sanitarios.

La obra se compone por una planta de tratamiento que es aplicada con alta tecnología y control informático, contará con 29 tanques de reservas y una red de distribución que comprenden tuberías de acero y PVC con una extensión de 152 kilómetros.

Bajo esta premisa nace la necesidad de implementar un sistema de facturación para el Proyecto Pesillo-Imbabura, que sirva para el control del consumo y suministro de agua potable de la población de las comunidades, que permitirá llevar una correcta facturación y seguimiento de la información en forma masiva y contribuirá al desarrollo socioeconómico de la comunidad.

1.2 Descripción del problema

En la provincia de Imbabura en la zona rural del cantón Otavalo, existen deficiencias en el control del consumo y suministro de agua potable; bajo esta premisa nace la necesidad de implementar un sistema de facturación para la población de las comunidades Cotama, Ilúman, Peguche, San Antonio, Gonzáles Suárez, Espejo, Carabuela, San José de Quichinche, Selva Alegre, San Rafael, San Juan, San Pablo.

En la actualidad, el control del consumo del agua potable y la facturación de la misma se realiza de forma manual, sin considerar los riesgos y problemas que pudiera ocasionar el manipular la información de esta manera; de igual condición, al registrar la participación de los miembros de las comunidades en eventos particulares como mingas, cuotas para preservar el medio ambiente, mantenimiento en sectores de la comunidad o sanciones por inasistencia a reuniones.

Llevar una correcta facturación y el control de información masiva contribuirá al desarrollo socioeconómico de la comunidad y la correcta distribución del suministro y proporcionar una mejor apreciación del servicio a las entidades y clientes que necesitan ser atendidos con rapidez y eficiencia.

1.3 Objetivos

1.3.1 Objetivo general.

Analizar, diseñar e implementar un sistema de facturación para las comunidades del Proyecto Pesillo Imbabura.

1.3.2 Objetivos específicos.

- Determinar una manera eficaz de recopilar los datos que serán ingresados al sistema.
- Determinar perfiles de usuario para manejo del sistema.
- Analizar los requerimientos de los miembros que forman parte de las unidades administradoras de cada comunidad.
- Sintetizar los parámetros y variables que van a ser automatizados y sistematizados en función de los requerimientos obtenidos.
- Diseñar en función de la metodología RUP (Proceso Unificado Racional) los diagramas y diseños del sistema que permitirán dar el enfoque y la estructura del sistema.
- Realizar las pruebas necesarias con el fin de encontrar los errores y depurando cada proceso con el fin de mantener un sistema confiable y robusto.
- Implementar el sistema en un punto central donde se realizará las correspondientes capacitaciones del mismo.
- Capacitar a los usuarios que estarán involucrados en el manejo del sistema con el propósito de generar el conocimiento para el uso y manejo del mismo.

1.4 Justificación

El proyecto se realizó con el fin de desarrollar e instaurar un sistema de facturación que servirá para el cálculo del consumo de agua potable mensual y las necesidades de las comunidades involucradas en el proyecto Pesillo Imbabura.

El sistema se enfoca además en llevar el registro de multas, control de asistencias de reuniones y donaciones, éstas últimas sirven para preservar el impacto ambiental y realizar remodelaciones dentro de las comunidades.

Dado a que todo registro se lo realiza manualmente, el sistema consiente en mejorar el control de la información, la misma que será procesada con rapidez, fiabilidad y confianza desde el ingreso, procesamiento, hasta la entrega de la información, optimizando tiempo, recursos y generando satisfacción, bienestar a los ciudadanos de la comunidad al mantener la facturación del sistema mensualmente.

El proyecto está enfocado en atender las necesidades de cada comunidad generando un producto sobre el cual se realizará un análisis de indicadores cuantitativos; tiempo de respuesta, frecuencia de uso, consumo de recursos, volumen de datos, cantidad de facturas generadas, monto total ingresos.

En cuanto a los indicadores cualitativos, el sistema puede destacar la medición de percepción de satisfacción, control de calidad del producto resultante, fácil manejo, entendimiento del sistema y resultados positivos de la inversión.

Por lo expuesto, se obtuvo un acuerdo con los representantes de cada comunidad con la finalidad de implementar este sistema.

1.5 Alcance

Este proyecto se realiza con el objetivo de garantizar un mejor control de información en la facturación mensual del consumo de agua potable y demás necesidades para las comunidades del Proyecto Pesillo Imbabura, debido a que actualmente el registro y control del suministro del agua y las actividades que desarrollan éstas comunidades se realizan de forma manual, lo que ocasiona la pérdida de información e incongruencia en la recolección de datos.

Para el desarrollo del sistema se tomará en cuenta los siguientes procesos:

1. Realizar un levantamiento de requerimientos de los usuarios en sitio.
2. Analizar y sintetizar los requerimientos obteniendo actividades automatizables.
3. Calcular el consumo de agua al mes teniendo en cuenta las multas y deudas de cada cliente.
4. Generación de reportes y facturas para usuarios finales.
5. Generación y determinación de claves para cada usuario que interactuará con el sistema dependiendo de sus tareas.
6. Diseñar el sistema en base a una metodología de desarrollo.
7. Desarrollar el sistema bajo estándares y normas.
8. Realizar pruebas de funcionamiento.

Cada módulo realizará las siguientes actividades:

1.6 Metodología

Durante el proceso de selección de la metodología a utilizar para el presente proyecto, se ha evaluado las etapas, roles, tiempos y resultados a obtener.

El análisis se basó según la tabla 1, en la cual se puede evidenciar las etapas y procesos.

Tabla 1. Metodologías existentes

	RUP	SCRUM	XP
Enfoque	Iterativo	Iterativo	Iterativo e incremental
Ciclo	Ciclo formal se define a través de 4 fases, pero algunos flujos de trabajo pueden ser concurrentes.	Cada sprint (iteración) es un ciclo completo.	Pruebas unitarias, continuas, frecuentemente repetidas y automatizadas
Planificación	Plan de proyecto formal, asociada a múltiples iteraciones, se utiliza. El plan es impulsado fecha final y también cuenta con hitos intermedios.	No de extremo a extremo del plan del proyecto. Cada plan de la siguiente iteración se determina al final de la iteración actual (no la fecha final de tracción). Dueño del Producto (usuario de negocios clave) determina el momento en que el proyecto se lleva a cabo.	Se realiza en un proceso de “Programación por parejas” que contiene una interacción con el usuario final, adicional una propiedad y simplicidad del código y reutilización del mismo.

continúa...

Tabla 1. Metodologías existentes (continuación...)

Alcance	Ámbito de aplicación está predefinido antes del inicio del proyecto y se documenta en el documento de Alcance. Ámbito de aplicación pueden ser revisados durante el proyecto, los requisitos se están aclarando, pero estas modificaciones están sujetas a un procedimiento estrictamente controlado.	En vez de alcance, SCRUM utiliza una Cartera de Proyectos, que se re-evaluado al final de cada iteración (sprint),	Utilizado para proyecto pequeño debido a la dificultad en la determinación de los costos del proyecto.
Los artefactos	Visión / Ámbito de aplicación del documento, el paquete formal de requisitos funcionales, documento de arquitectura del sistema, plan de desarrollo, plan de pruebas, scripts de prueba, etc.	El único artefacto formal es el software operativo.	Se realiza un proceso de roles o división de procesos los cuales son (Programador, cliente, coaching, consultor, gestor, testeo)
Tipo de proyecto / producto	Recomendado para grandes, a largo plazo, a nivel de empresa con proyectos a medio y alta complejidad.	Recomendado para las mejoras rápidas y organizaciones que no dependen de una fecha límite.	Utilizado para proyectos con alto grado de riesgos, definido con un grupo pequeño de programadores entre 2 y 12

Fuente: (Sánchez, 2003)

Este proyecto se desarrollará bajo un la metodología RUP (Rational unified Process), debido a que es un proceso incremental que esta guiado por casos de uso, en donde un usuario puede desempeñar distintos roles o perfiles durante el proceso. La metodología basada en este sistema muestra el proceso de desarrollo del sistema con pasos firmes y establecidos, incluye un conjunto de metodologías que van adaptándose al contexto y necesidades del cliente según se desenvuelva el mismo.

La estructura dinámica de RUP permite que el sistema sea un proceso de desarrollo iterativo, inmerso en cuatro fases: Inicio, elaboración, desarrollo y cierre.

La metodología ampliada en este proyecto proporciona ventajas a las tareas que se ejecutan en cada uno de los módulos; estima un mejor manejo de las actividades y procesos, midiendo la velocidad de las interacciones sobre la arquitectura del software y la puesta en práctica del mismo.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Proceso de recaudación

Las comunidades del proyecto Pesillo Imbabura realizan la facturación del agua potable en base al registro de consumo.

2.1.1 Descripción del proceso.

El proceso de recaudación se realizaba de una forma empírica, señalando como primera observación el no control y falta de criterio al realizar el registro de consumo.

La persona asignada por la comunidad acudía a cada domicilio que tenía el servicio de agua o riego, quien registraba la métrica actual manual del consumo en el momento y realizaba el mismo proceso con todas las casas del sector asignado.

Una vez recopilada esta información se realizaba el ingreso de datos mediante la aplicación Excel y se realizaba una resta con el consumo anterior y el valor obtenido se multiplicaba por la cifra acordada del consumo de m³ de agua y se obtiene el valor a pagar el mismo que era mencionado a cada suscriptor para su cancelación.

El proceso de multas no se llevaba un control, estos valores no tenían la forma de ser auditados por tal motivo los valores eran de difícil control.

Los eventos no tenían la forma de comunicación hacia los habitantes del sector por tal motivo la participación era muy baja por falta de un control de las personas necesarias para la realización del mismo.

El control de medidores y asignación del mismo era sin mayor registro por ende existe personas que tienen una mayor cantidad de medidores a comparación de habitantes.

Todos estos aspectos fueron evaluados con las juntas de agua de cada comunidad con el fin de obtener procesos similares para la creación de los módulos respectivos.

2.2 Herramientas y tecnología

Las herramientas y tecnologías seleccionadas para la creación del sistema se describen a continuación:

2.2.1 Gestores de programación.

- **Java**

Java es un lenguaje de programación que permite desarrollar aplicaciones útiles y de fácil uso en sistemas operativos por ejemplo Windows, Linux, Apple.

- **Javascript**

JAVA Script es un lenguaje orientado a eventos acoplado al manejo de datos convirtiéndolo en un sistema multiplataforma, permitiendo de esta manera ser un sistema cliente servidor.

- **JDK (Java Development Kit)**

Es un conjunto de herramientas, programas y librerías que desarrollan, compilan, ejecutan y generan documentación en el lenguaje Java, es decir permite la traducción del código de programación a un lenguaje de usuario.

- **JSP (Java Server Pages)**

-

Es una tecnología orientada a crear páginas web con programación Java, permite crear

aplicaciones que se ejecuten en variados servidores web de múltiples plataformas. En el sistema; permite la interconexión de códigos de programación en un sistema multicapa.

- **JSF (Fava Server Faces)**

Java Server Faces es un framework o ambiente de trabajo, destinado a facilitar la construcción y el desarrollo de interfaces para aplicaciones basadas en entornos Web para el sistema permite visualizarlo mediante el servicio de Apache.

- **XMLI**

Extensible Markup Language, es un lenguaje diseñado especialmente para los documentos de la web. Permite crear las formas y diseño fijo del sistema.

2.2.2 Gestores de base de datos.

Base de datos es un conjunto exhaustivo no redundante de datos estructurados y organizados y la implementación en máquinas accesibles en tiempo real los mismos que son compatibles con usuarios concurrentes que necesitan de cierta información.

- **SQL (Structured Query Language)**

Es un lenguaje para base de datos estandarizados, que interactúan al usuario con el sistema mediante sentencias definidas.

- **PostgresSQL**

Es un potente sistema de base de datos objeto-relacional de código, permite una interrelación con los sistemas de programación.

2.2.3 Lenguaje de programación.

Para este proyecto se ha visto necesario utilizar JSP (Java Server Pages), relacionado con código XML para crear un diseño fijo y relacionado, adicionalmente se conecta con una base de datos que permite al usuario realizar las consultas mediante un modo grafico dejando atrás el texto tradicional, por tal motivo se ha aplicado sobre un servidor web con el fin de tener disponibilidad en un entorno cliente – servidor, que potencialmente puede ser alojado en un servidor aparte con fines de integración

2.3 Generalidades

2.3.1 Metodología.

2.3.1.1 RUP (Rational Unified Process).

Es un conjunto de metodologías que se adaptan al contexto y necesidades de cada organización y que junto al lenguaje UML, constituye una metodología estándar que es usada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Este tipo de modelo permite el desarrollo de software a gran escala siguiendo un proceso continuo de retroalimentación y pruebas, garantizando el cumplimiento de muchos estándares de calidad que genera mayor complejidad en los controles administrativos, y sus beneficios recompensan el esfuerzo invertido.

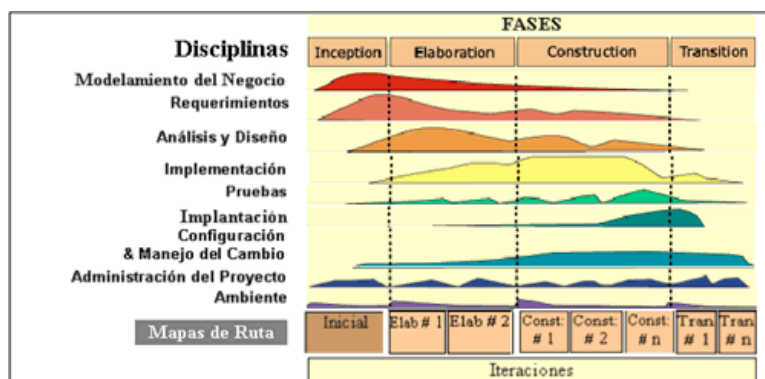
Ciclo de vida del RUP

Esta metodología se divide en 4 fases que son:

- **Fase de concepción o iniciación:** En esta fase se define y establece el alcance del proyecto proponiendo una visión general de la arquitectura del software.

- **Fase de elaboración:** En esta fase se selecciona los casos de uso para definir la base del sistema llegando a una solución preliminar.
- **Fase de construcción:** En esta fase se completa la funcionalidad del sistema tomando en cuenta todos los requerimientos que se vayan dando en el desarrollo del proyecto.
- **Fase de transición:** En esta fase se asegura la disponibilidad del sistema a los usuarios finales a través de capacitación, verificación y ajustes de errores en la ejecución de pruebas.

Figura 1. Ciclo de vida del RUP



Fuente:(Scott W. Amber, 2014)

Para aplicar este tipo de metodología se recomienda conocimiento de otros lineamientos, el desarrollo de habilidades de los analistas, programadores, DBAs, y demás miembros del equipo de desarrollo para conocer el enfoque de RUP, ya que muchas de las veces esta metodología no es aplicada por desconocimiento de la misma.

Principios claves en los que está basado el RUP:

Adaptar el proceso: El proceso debe adaptarse al cliente en base a sus necesidades, debido a que es muy importante interactuar con él. En el diseño influyen las condiciones y regulaciones que se especifican al inicio.

Equilibrar prioridades: Es indispensable encontrar el equilibrio que permita abarcar los requisitos y las contradicciones de los diversos participantes con el fin de evitar contrariedades a futuro.

Demostrar valor iterativamente: Se menciona que en cada iteración los proyectos se entregan en un modo de etapas iteradas. En cada iteración se analiza, la estabilidad y calidad del producto, y los riesgos del proyecto por cada uno de los involucrados.

Colaboración entre equipos: Debe existir una comunicación fluida entre todas las personas del equipo para coordinar requisitos, desarrollo, evaluaciones, planes, resultados.

Elevar el nivel de abstracción: Evita que los ingenieros de software partan directamente de los requisitos a la codificación de software, sin saber con certeza qué codificar para satisfacer de la mejor manera los requisitos y sin comenzar desde un principio considerando la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

Enfocarse en la calidad: El aseguramiento de la calidad depende del proceso de desarrollo y debe manifestarse no solo al final de cada iteración, sino en todos los aspectos de la producción.

Roles de la Metodología RUP

Figura 2. Roles de la metodología RUP



Fuente: (Scott W. Amber, 2014)

Cada fase en RUP sus iteraciones pueden descomponerse. Un ciclo de desarrollo completo es una iteración proporcionando como resultado una entrega de producto ejecutable (interna o externa).

El proceso consiste en una serie de roles, cada uno de estos roles son otorgados a los miembros o contribuidores del proyecto, la misma que delega las tareas y el resultado a cada uno, por ende RUP en sus ciclos de desarrollo realiza una serie de acciones necesarias para el análisis, implementación y diseño del sistema.

2.4 Diagramas UML

UML es un lenguaje unificado de modelado de sistemas de software el mismo que permite modelar, construir, visualizar y documentar un sistema al igual que describir métodos o procesos para el mismo.

No se trata de un lenguaje de programación ya que usa una interfaz gráfica en donde solo se permite diagramar cualquier tipo de representación a un requerimiento.

Existen varios diagramas UML los mismos que se complementan para obtener mejores resultados al momento de empezar a trabajar en el proyecto. Dentro de estos se han

considerado los siguientes:

- Diagrama de casos de uso
- Diagrama de actividades
- Diagrama de secuencia
- Diagrama de clases

2.4.1 Diagramas caso de uso.

Representan las funciones y comportamiento del sistema; puede ejecutarse desde el punto de vista del usuario, con esto se puede interpretar fácilmente los aspectos importantes que desempeña el mismo sin usar contextos complejos.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

Este capítulo describe el análisis y diseño de los diferentes módulos, diagramas utilizados durante la elaboración del sistema.

3.1 Recolección de requisitos y necesidades

La necesidad del sistema empezó al momento de realizar la toma de los requerimientos de las comunidades involucradas determinando los aspectos globales para realizar un sistema completo, lógico y de fácil aprendizaje hacia el usuario.

La recolección de requerimientos se la realizó en Otavalo, el 15 de noviembre del 2012, en conjunto con el Ingeniero Daniel Díaz y los representantes de las comunidades participantes del proyecto.

En base a los requerimientos globales se determinó lo siguiente:

- Manejo de los valores de acuerdo a la comunidad.
- Ingreso de N registros de usuarios.
- Ingreso de campo Multas con el fin de registrar valores por pagar por obras realizadas e inasistencias.
- Ingreso de registro de actividades de la comunidad.
- Ingreso de comunidades y sectores respectivamente.
- Módulo de creación de perfiles de acuerdo al usuario.
- Registro de habitantes por sector.

3.2 Proceso de registro

El proceso de registro, procesamiento, facturación del sistema se realiza de la siguiente forma:

- El lector ingresa los datos en el procesador de texto el cual fue previamente realizado en base a la lectura de cada medidor del sector con su respectivo número de suministro.
- Los datos ingresados en el procesador de texto deben constar del número de suministro y la lectura actual, este archivo debe ser guardado en un sector del disco C del equipo.
- Una vez copiado el archivo en el disco C, el administrador o encargado de la tarea de lecturas debe ingresar al sistema e ingresar al módulo de carga de información.
- Una vez registrada la ruta de la ubicación del archivo inicialmente, el administrador debe realizar un clic en el botón “CARGAR”, este botón contiene una función que realiza una carga de información masiva a una tabla temporal que a su vez guarda en la Base de datos las nuevas lecturas.
- Una vez realizado este proceso, el administrador del sistema debe crear los usuarios respectivos para el uso de la aplicación.
- Al momento que un cliente desee cancelar su planilla de pago debe mencionar el código de su medidor con el fin de validar los datos del cliente.
- Cabe señalar que el sistema ha sido parametrizado con el fin de optimizar los tiempos de gestión.

3.3 Módulos del sistema de facturación

Para construir un software eficiente con características que permitan la flexibilidad para un futuro escalamiento de requerimientos se lo ha dividido en los siguientes módulos:

Figura 3. Módulos del sistema de facturación



Elaborado por: Livardi Salgado & Daniel Sierra

3.3.1 Módulo de administrador.

Esté módulo brindará la capacidad de gestionar y administrar los usuarios, perfiles y accesos al sistema contando con los siguientes procesos:

- **Gestión de Usuarios:** Donde se creará y editará los usuarios que van acceder al sistema.
- **Gestión de Perfiles:** En este proceso se podrá crear, editar y asignar los perfiles de usuario y permisos que cada usuario tendrá en el sistema.
- **Autenticación:** Podrá solicitar la re-asignación de contraseñas cuando el

usuario lo requiera.

- **Gestión de Parámetros:** En este proceso se podrá asignar los valores de los parámetros que son regulados por cada comunidad (Por ejemplo: Costo por cm³, costo de básico, etc.).
- **Auditoria del sistema:** En este proceso se realizará las pruebas de caja negra, la misma detectará las falencias del sistema por alguna entidad o personal que no pertenezca al grupo creador del software.

3.3.2 Módulo de registro.

Este módulo es esencial para registrar a los clientes, ingresando los datos personales, de la comunidad, del medidor, etc.; además se registrará las diferentes lecturas tomadas de los medidores.

Tabla 2. Nivel de gestión de clientes

NIVEL 1	GERENCIAL	Permisos exclusivos
	TECNOLOGÍA	Permisos de administrador
	DIRECCIONES	Permisos de acuerdo al área
NIVEL 2	ASISTENTES	Permisos de acuerdo a sus labores entregados
	CAJEROS	Permisos de facturación y manipulación
NIVEL 3	OPERADORES	Permisos para entrega y digitación de información recibida de la lectura
	POSICIONES BÁSICAS	Permisos restringidos de información

Elaborado por: Livardi Salgado & Daniel Sierra

Contará con los siguientes procesos:

Gestión de clientes: Donde se podrá realizar las siguientes tareas:

- a. Creación de clientes.
- b. Modificación de clientes.
- c. Consulta por cliente, comunidad y medidor.
- d. Eliminación de clientes.

Forma de ingreso:

La forma de ingreso será manual y su identificador es el número de medidor el mismo que permite enlazar con la información del suscriptor con el fin de obtener un proceso de facturación fiable.

Se ha determinado las siguientes formas de registro:

- Código de medidor

Al ingreso de datos se puede validar la comprobación de información:

- Nombre del cliente
- Última medición
- Dirección o predio

A nivel de registro, la categorización puede variar de acuerdo a las siguientes necesidades:

Tabla 3. Niveles de registro

NIVEL 1	CÓDIGO O SERIAL ÚNICO DE MEDIDOR	Es el código del medidor el mismo que permite conocer la información total del estado del servicio, detalle de su propietario, valores a cancelar, valores pendientes y un resumen histórico de su uso.
NIVEL 2	CÉDULA DE IDENTIDAD DEL SUSCRIPTOR	Es el Código Único del suscriptor que permite el registro de medidor.

Elaborado por: Livardi Salgado & Daniel Sierra

3.3.3 Módulo de cobro y facturación.

En este módulo el usuario asignado realizará el cobro del pago por cada cliente, ingresando el número de medidor.

El sistema responderá con el monto total a pagar incluido interés por mora y sanciones aplicadas, dando la opción de realizar pagos diferidos una vez que el administrador del sistema le haya otorgado el permiso correspondiente a la persona que factura al cliente.

En este módulo realizará las siguientes tareas:

- Consulta de planilla a pagar.
- Impresión de factura.
- Ingreso de pagos diferidos.
- Consulta de ingreso diario.

Se detallan las formas de consultas posibles es decir, los identificadores de búsqueda que permitan recibir información desde la base de datos.

Forma de pago:

Se ha determinado las siguientes formas de pago:

- Un solo pago
- Diferido

Los usuarios pueden tener la siguiente clasificación de facturas:

- Factura cobrada
- Factura actual
- Factura atrasada o en mora

A nivel de facturación, la categorización de la siguiente forma puede variar dependiendo de las necesidades a presentarse:

Tabla 4. Niveles de facturación

NIVEL 1	ABONO O PAGO	Se identifica cuando la factura ha sido cancelada dentro del tiempo pre – establecido.
NIVEL 2	FACTURACIÓN PENDIENTE	Se identifica cuando la factura tiene mora pero la misma no tiene otra que le antecede, conocido como doble factura
NIVEL 3	FACTURACIÓN EN MORA	Se identifica cuando el suscriptor registra 2 o más facturas pendientes de pago

Elaborado por: Livardi Salgado & Daniel Sierra

Las facturas serán emitidas y cobradas de acuerdo a la fecha de medición del servicio de agua potable.

3.3.4 Módulo de reportes.

El módulo de reportes será capaz de realizar consultas bajo filtros de búsqueda ya sea por: número de planilla, número de medidor, cédula de identidad y realizará las siguientes tareas:

- Consulta de reportes
- Impresión de reporte
- Exportación de reportes a archivos formato Excel o .pdf.

Perfiles de usuario:

Se ha determinado los siguientes tipos de perfiles:

- Registrador
- Auditor de reportes
- Facturador

Los usuarios pueden tener la siguiente clasificación de permisos:

- Lectura
- Escritura
- Totales(administración)

3.4 Diagramas

Se presenta la diagramación de todas las interfaces y funcionalidades del sistemas tanto en el modo de uso, la diagramación de secuencia y actividad de cada a evento a desarrollar

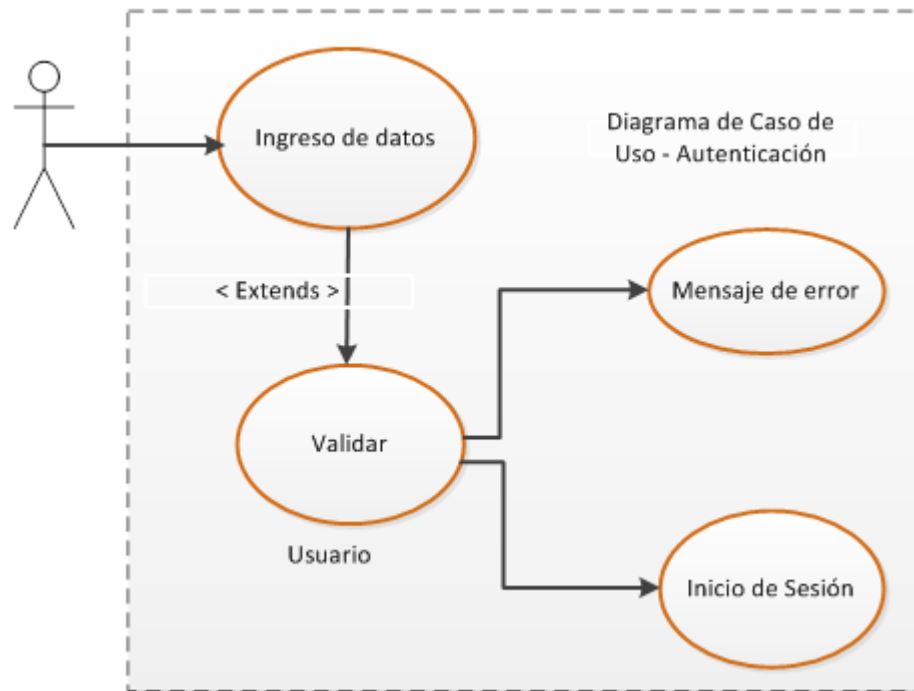
3.4.1 Modulo Autenticacion.

En el presente modulo se puede identificar el acceso al sistema y sus autorizadores.

3.4.1.1 Proceso autenticación.

Modelo de Caso de Uso #1.

Figura 4. Modelo de Caso de uso #1, Proceso Autenticación



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #1.

Tabla 5. Escenario de Caso de Uso, Proceso Autenticación

ACTORES	Usuario Administrador
PRECONDICIONES	El usuario debe estar ingresado al sistema.
POSCONDICIONES	El usuario ingresará al sistema para realizar sus tareas correspondientes según los permisos del administrador. El Administrador ingresará sin ningún tipo de restricción.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El usuario accede al sistema ingresa sus credenciales: Usuario y Contraseña, y da clic en ingresar2. El sistema toma estos datos y lo valida con la base de datos.3. El sistema inicia la sesión del usuario logueado
FLUJO ALTERNATIVO	El flujo básico debe ser de la siguiente forma: 2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos

Elaborado por: Livardi Salgado & Daniel Sierra

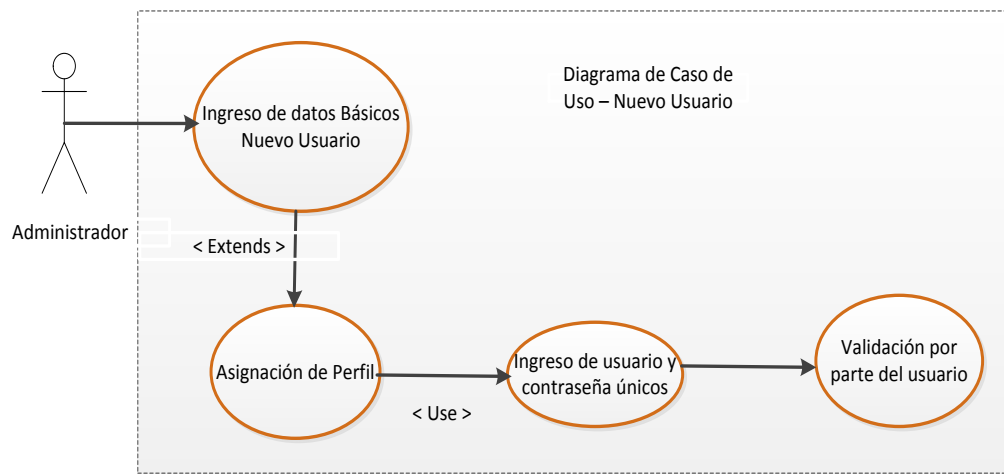
3.4.2 Modelo usuario.

3.4.2.1 Proceso creación de usuario.

Se define a la creación de los diferentes usuarios que ejecuten y procesen la información del sistema (colaboradores).

Modelo de Caso de Uso #2.

Figura 5. Modelo de Caso de uso, Proceso Creación de Usuario.



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #2.

Tabla 6. Escenario de Caso de Uso, Proceso Creación de Usuario

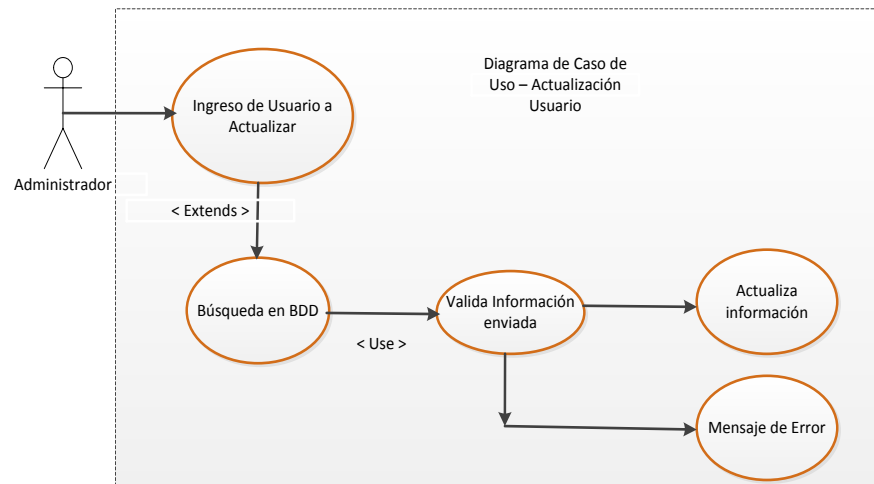
ACTORES	Administrador Usuario
PRECONDICIONES	Predefinir sus actividades a realizar para evaluar el perfil.
POSCONDICIONES	El usuario y/o administrador realizará la validación de información para la asignación del perfil adecuado.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El administrador recolecta los datos necesarios.2. El administrador ingresa todos los datos obtenidos y genera el usuario y certifica el inicio de sesión del usuario logueado.
FLUJO ALTERNATIVO	El flujo básico debe ser de la siguiente forma: 2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos o no válidos.

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.2.2 Proceso actualización de usuario.

Modelo de Caso de Uso #3.

Figura 6. Modelo de Caso de uso, Proceso Actualización de Usuario

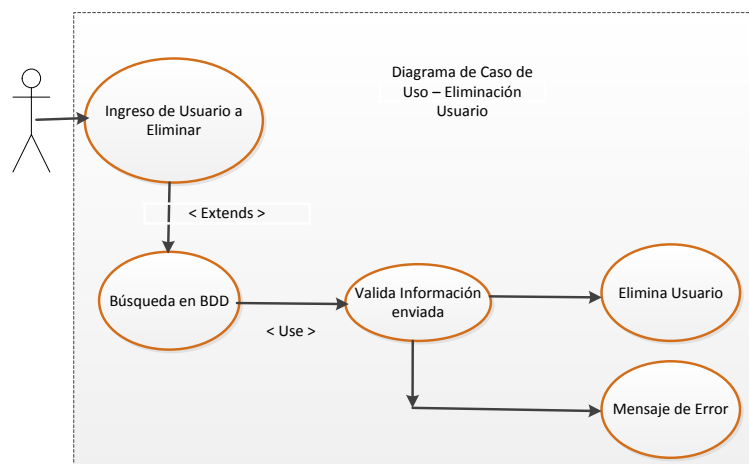


Elaborado por: Livardi Salgado & Daniel Sierra

3.4.2.3 Proceso eliminación de usuario.

Modelo de Caso de Uso #4.

Figura 7. Modelo de Caso de uso, Proceso Eliminación de Usuario



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #4.

Tabla 7 Escenario de Caso de Uso, Proceso Eliminación de Usuario

ACTORES	Administrador Usuario
PRECONDICIONES	Verificación de funcionalidades del usuario
POSCONDICIONES	Retiro del perfil
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El administrador busca al usuario a eliminar2. Una vez identificado el usuario el mismo se retira perfiles y se elimina del sistema.
FLUJO ALTERNATIVO	El flujo básico debe ser de la siguiente forma: 2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos o no encontrados.

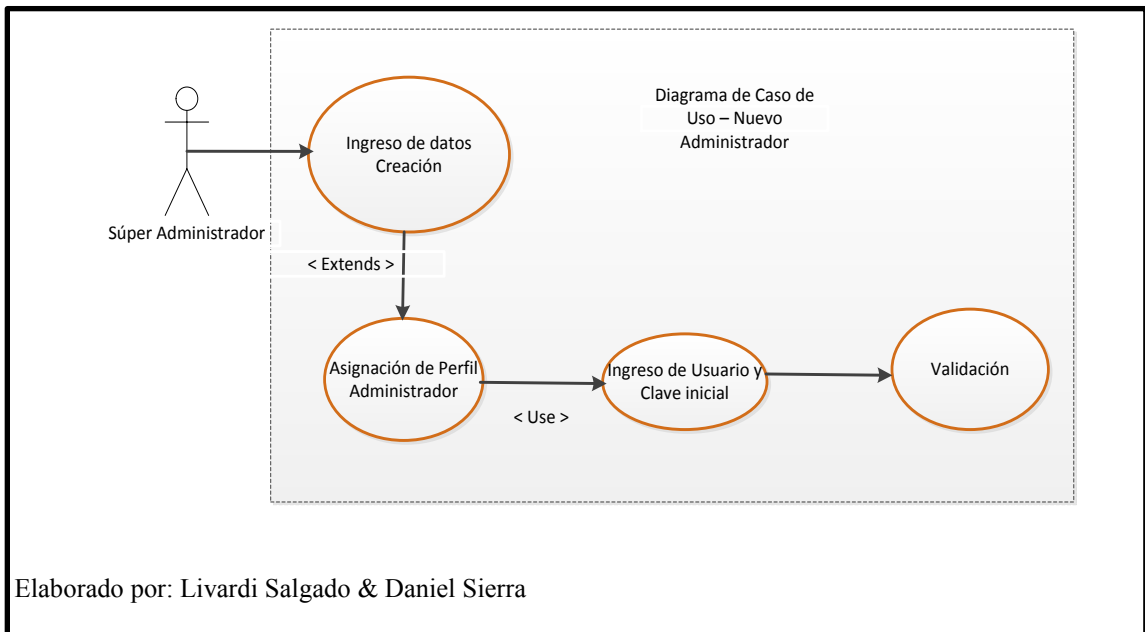
Elaborado por: Livardi Salgado & Daniel Sierra

3.4.3 Modelo administrador.

3.4.3.1 Proceso creación.

Modelo de Caso de Uso #5.

Figura 8. Modelo de Caso de uso, Proceso Creación de Administrador.



Escenario de Caso de Uso #5.

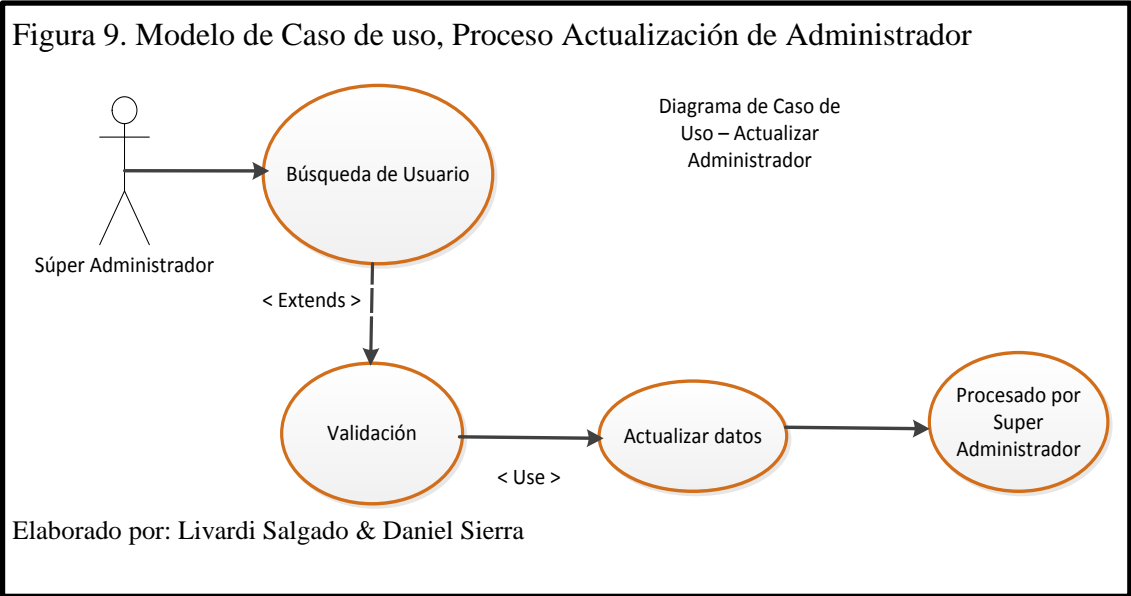
Tabla 8. Escenario de Caso de Uso, Proceso Creación de Administrador

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Predefinir sus actividades a realizar para Asignar Perfil.
POSCONDICIONES	El Súper administrador realizará la validación de información necesaria.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El súper administrador crea un administrador de sistema con los datos recolectados.2. El administrador creado valida su acceso y sus funcionalidades en el sistema.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos o no válidos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.3.2 Proceso actualización de administrador.

Modelo de Caso de Uso #6.



Escenario de Caso de Uso #6.

Tabla 9. Escenario de Caso de Uso, Proceso Actualización de Administrador

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Considerar Perfil.
POSCONDICIONES	El Súper administrador realizará la Actualización.
	El flujo básico debe ser de la siguiente forma:

continúa...

Tabla 9. Escenario de Caso de Uso (continuación...)

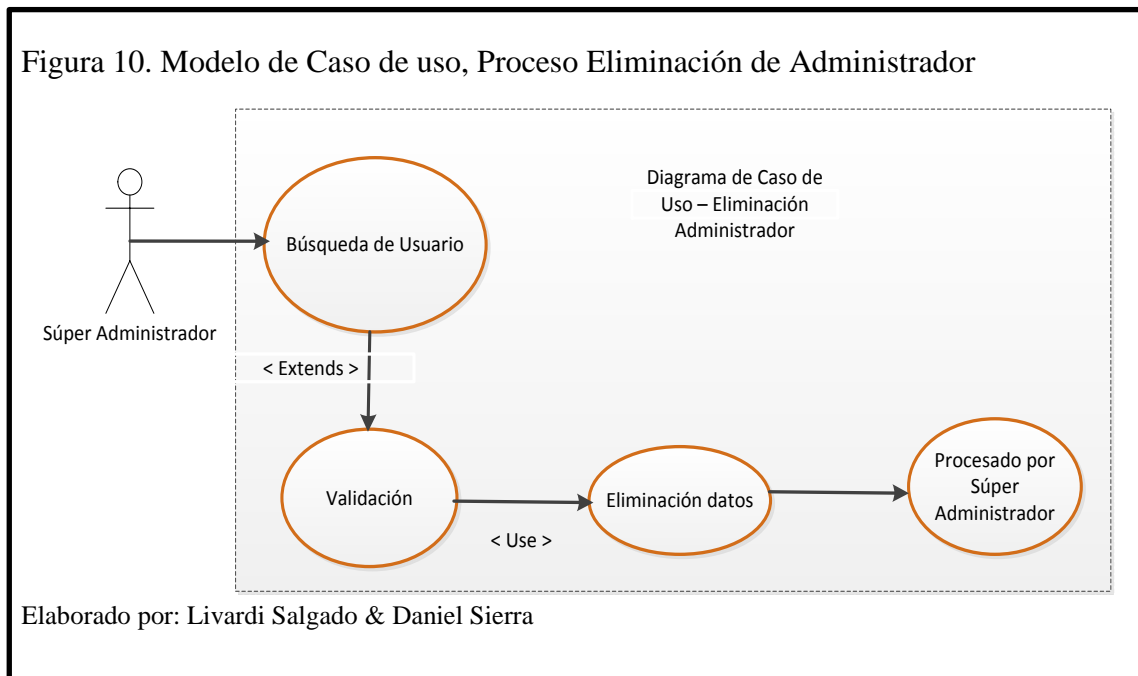
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El súper administrador realiza la búsqueda del usuario a realizar la actualización. 2. El súper administrador actualiza la información necesaria del administrador. 3. El administrador actualizado confirma la información.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.3.3 Proceso eliminación de administrador.

Modelo de Caso de Uso #7.

Figura 10. Modelo de Caso de uso, Proceso Eliminación de Administrador



Escenario de Caso de Uso #7.

Tabla 10. Escenario de Caso de Uso, Proceso Creación de Administrador

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Análisis de Perfil.
POSCONDICIONES	El Súper administrador realizará la validación de información necesaria.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El súper administrador realiza la búsqueda del administrador.2. Una vez localizado el usuario se retira los perfiles y se elimina el registro del usuario.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos y no procesa ninguna eliminación.</p>

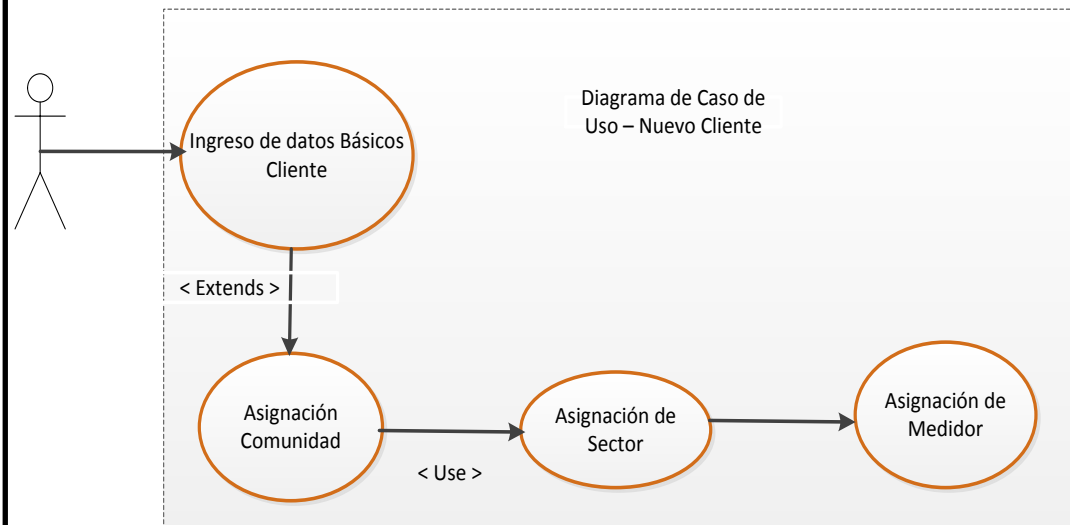
Elaborado por: Livardi Salgado & Daniel Sierra

3.4.4 Modelo clientes.

3.4.4.1 Proceso creación.

Modelo de Caso de Uso #8.

Figura 11. Modelo de Caso de uso, Proceso Creación de Clientes



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #8 – Creación de Clientes.

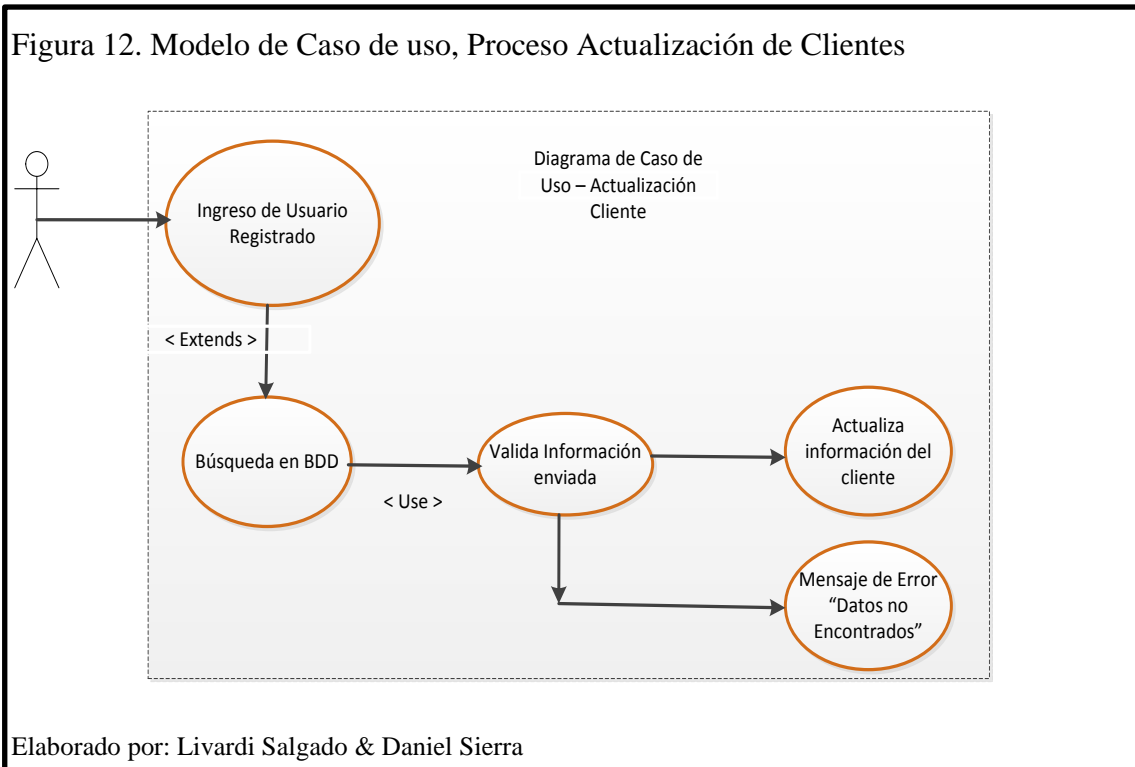
Tabla 11. Escenario de Caso de Uso, Proceso Creación de Clientes

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Recolección de datos personales
POSCONDICIONES	Cliente registrado.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El administrador o usuario con perfil adecuado obtiene la información de los clientes y verifica los datos.2. Una vez realizada la verificación se ingresa la información del cliente.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.4.2 Proceso actualización.

Modelo de Caso de Uso #9.



Escenario de Caso de Uso #9.

Tabla 12. Escenario de Caso de Uso, Proceso Actualización de Clientes

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Cliente debe estar registrado.
POSCONDICIONES	Cliente actualiza datos necesarios.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: 1. El usuario autorizado o administrador realiza la búsqueda del cliente mediante su identificador.

continúa...

Tabla 12. Escenario de Caso de Uso (continuación...)

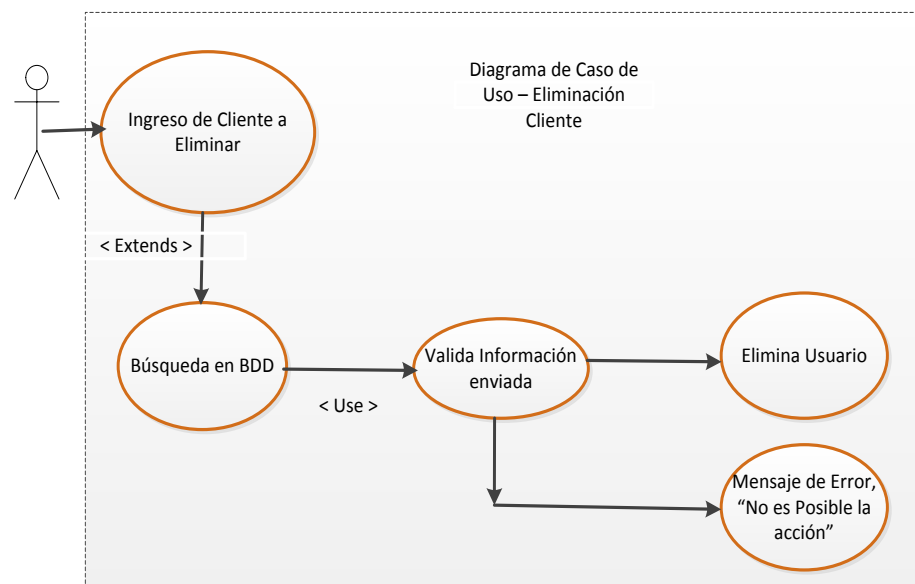
	2. Una vez localizado el cliente en el sistema el administrador o usuario autorizado realiza la actualización solicitada.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.4.3 Proceso de eliminación de clientes.

Modelo de Caso de Uso #10.

Figura 13. Modelo de Caso de uso, Proceso Eliminación de Clientes



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #10.

Tabla 13. Escenario de Caso de Uso, Proceso Creación de Administrador

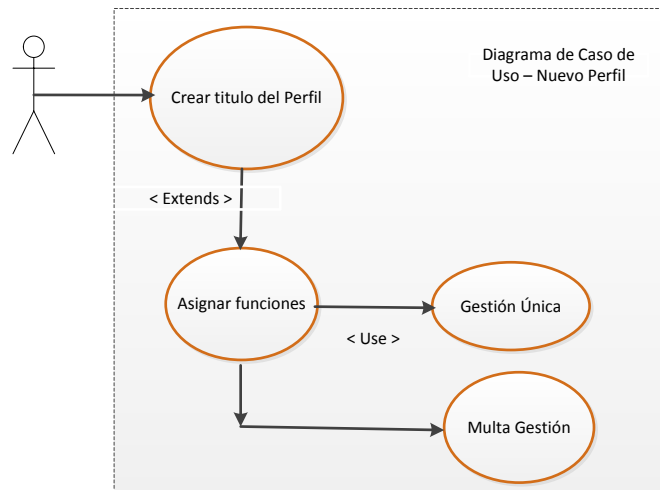
ACTORES	Súper Administrador Administrador
PRECONDICIONES	Predefinir sus actividades a realizar para Asignar Perfil.
POSCONDICIONES	El Súper administrador realizará la validación de información necesaria.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la búsqueda del cliente mediante su identificador.2. Una vez localizado el cliente en el sistema el administrador o usuario autorizado realiza la eliminación analizando que no tenga valores pendientes de pago.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos o no válidos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.5 Modelo perfiles.

Modelo de Caso de Uso #11.

Figura 14. Modelo de Caso de uso, Proceso Creación de Perfil



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #11.

Tabla 14. Escenario de Caso de Uso, Proceso Creación de Perfil

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Predefinir funciones del Perfil
POSCONDICIONES	Análisis de conformidad por parte del nuevo usuario de las funcionalidades otorgadas validando su función.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El administrador identifica la necesidad de crear un nuevo perfil.2. Genera un nombre de perfil y asigna las diferentes

continúa...

Tabla 14. Escenario de Caso de Uso (continuación...)

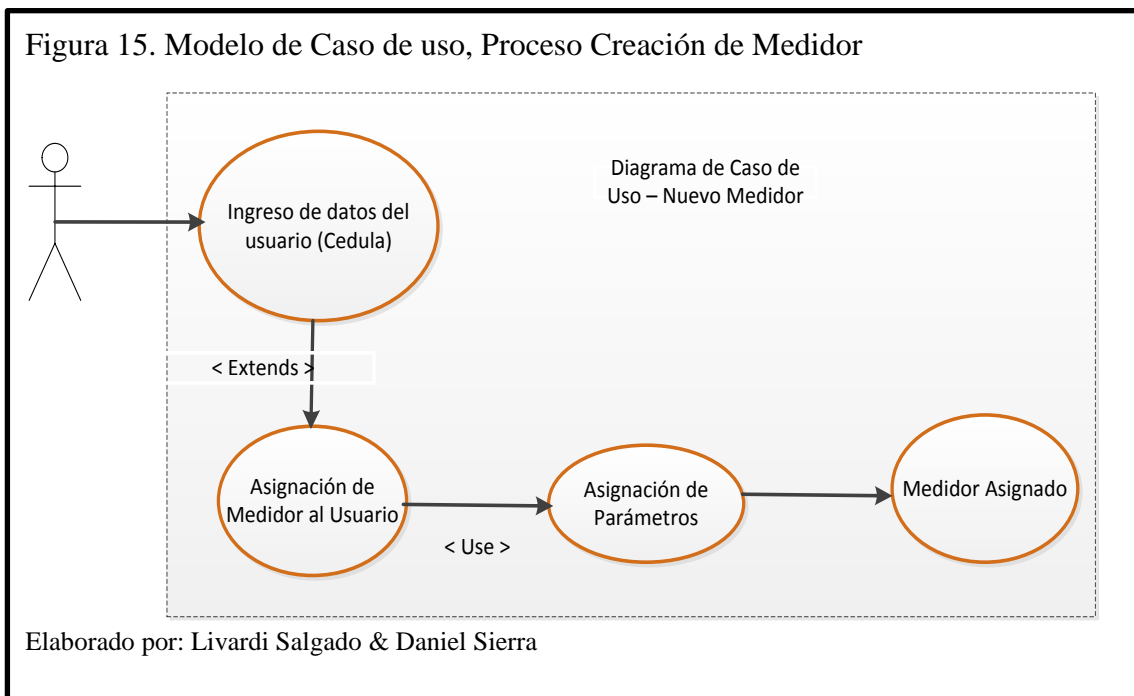
	funcionalidades para cada perfil, estas pueden ser incluso multifunción.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si la funcionalidad del perfil no es adecuada se inicia el proceso de creación.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.6 Modelo medidor.

3.4.6.1 Proceso creación.

Modelo de Caso de Uso #12.



Escenario de Caso de Uso #12.

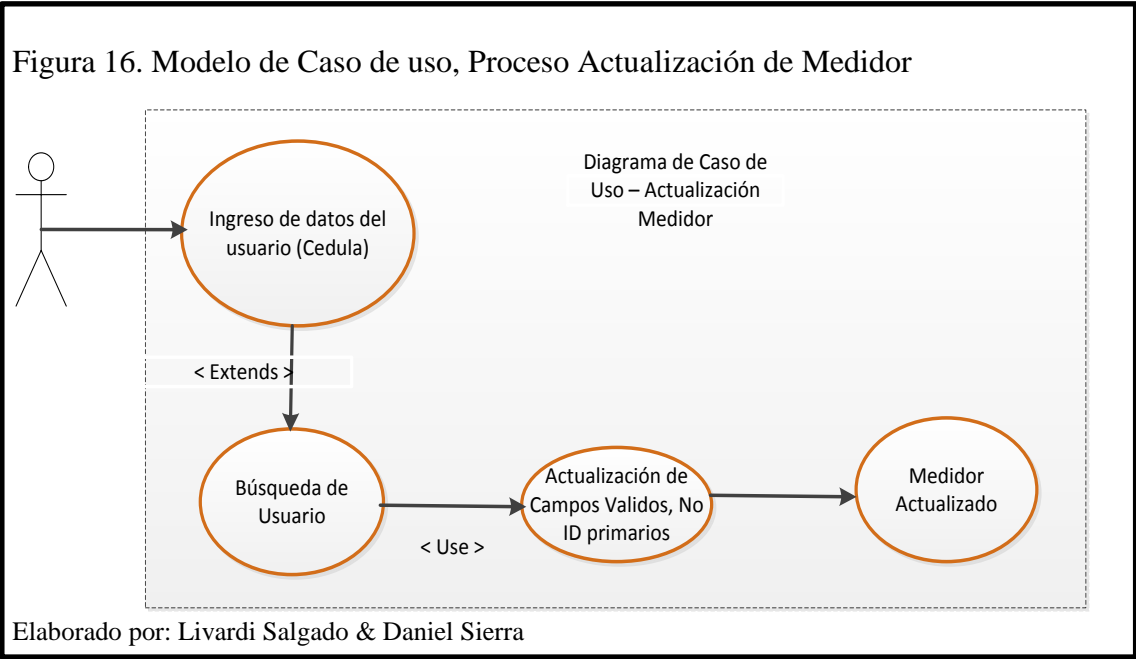
Tabla 15. Escenario de Caso de Uso, Proceso Creación de medidor

ACTORES	Administrador Usuario
PRECONDICIONES	Haber ingresado al sistema y encontrarse dentro del módulo de registro de cliente.
POSCONDICIONES	El usuario y/o administrador realizará la validación de información para la asignación del medidor correspondiente para el cliente.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la asignación de un medidor a un cliente previamente creado.2. Una vez asignado se determina los parámetros de este nuevo medidor y se determina su contador inicial es vital este dato por el tema de medición de consumo.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.6.2 Proceso actualización.

Modelo de Caso de Uso #13.



Escenario de Caso de Uso #13.

Tabla 16. Escenario de Caso de Uso, Proceso Creación de Medidor

ACTORES	Súper Administrador Administrador Usuario autorizado
PRECONDICIONES	Validación de Medidor creado anteriormente.
POSCONDICIONES	Asignación de parámetros actualizados al Medidor.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: 1. El usuario autorizado o administrador realiza la

continúa...

Tabla 16. Escenario de Caso de Uso (continuación...)

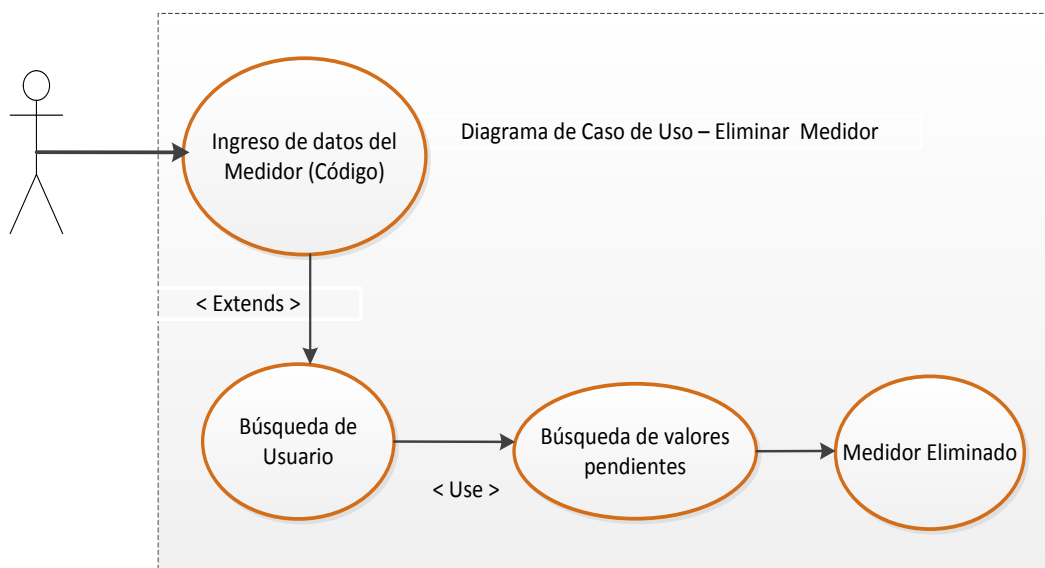
	<p>búsqueda del código de medidor.</p> <p>2. Una vez encontrado el medidor se habilitan los campos editables del medidor con el fin de actualizarlo.</p>
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos o no válidos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.6.3 Proceso eliminación.

Modelo de Caso de Uso #14.

Figura 17. Modelo de Caso de uso, Proceso Eliminación de Medidor



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #14.

Tabla 17. Escenario de Caso de Uso, Proceso Eliminación de Medidor

ACTORES	Súper Administrador Administrador Usuario autorizado.
PRECONDICIONES	Validar de información del medidor.
POSCONDICIONES	Eliminación de Medidor con cero valores.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la búsqueda del medidor mediante su identificador.2. Una vez localizado el medidor en el sistema el administrador o usuario autorizado realiza la eliminación solicitada.
FLUJO ALTERNATIVO	El flujo básico debe ser de la siguiente forma: 2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos

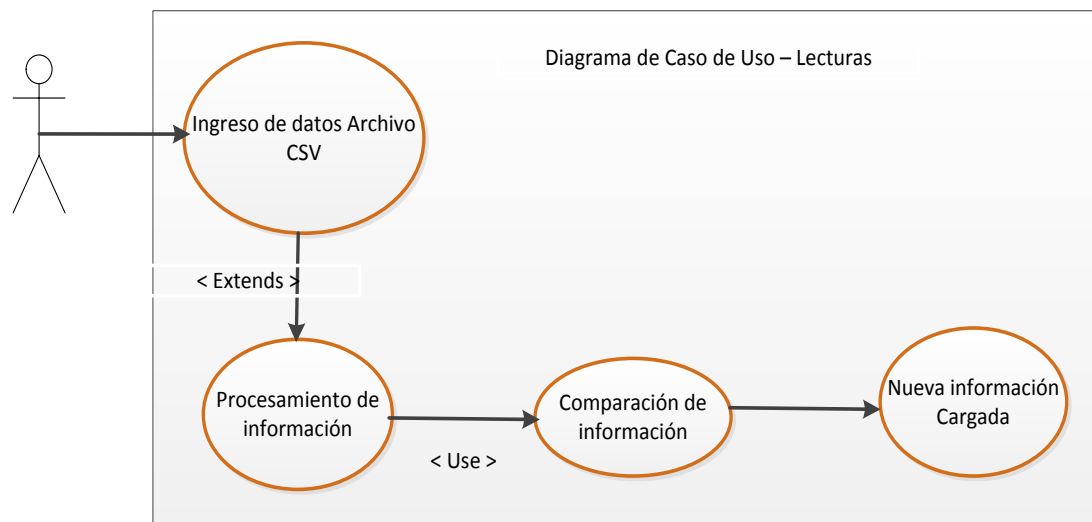
Elaborado por: Livardi Salgado & Daniel Sierra

3.4.7 Modelo lectura de información.

3.4.7.1 Proceso registro.

Modelo de Caso de Uso #15.

Figura 18. Modelo de Caso de uso, Proceso Lectura



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de Uso #15.

Tabla 18. Escenario de Caso de Uso, Proceso Lectura

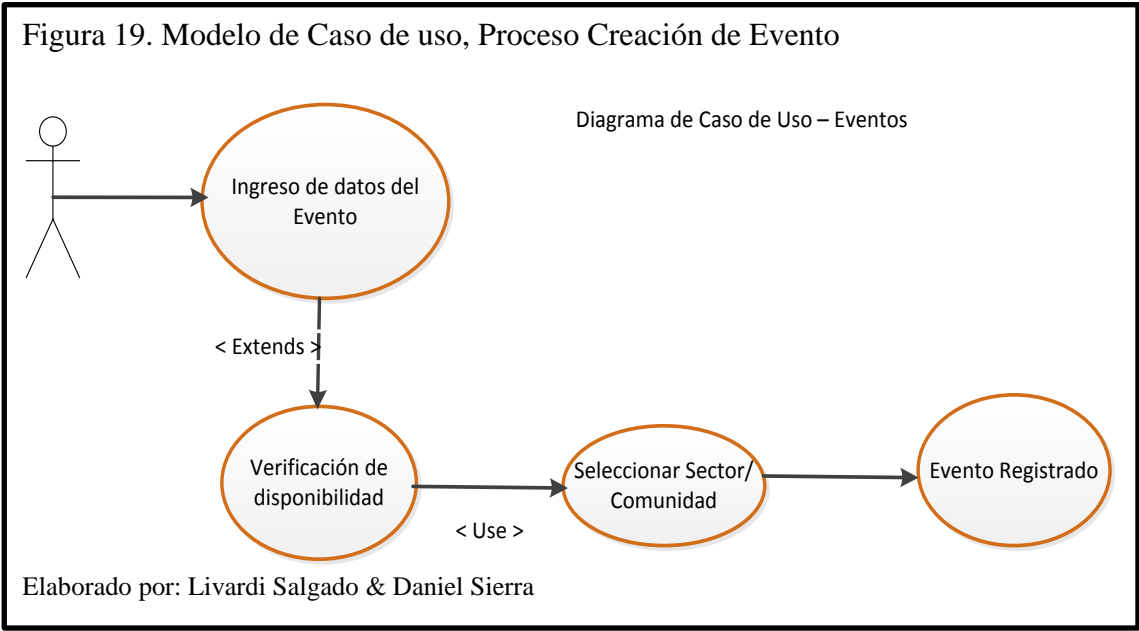
ACTORES	Súper Administrador Administrador Usuario
PRECONDICIONES	Verificación de archivo valido.
POSCONDICIONES	Registro de la información y carga a las tablas de Base de Datos.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la carga de la información mediante un archivo en formato csv previamente realizado.2. La aplicación recibe la información mediante el módulo de carga y la misma recibe los datos del contenido y es actualizado en la tabla de la base de datos.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos no coinciden el sistema presenta un mensaje indicando que los datos ingresados son incorrectos.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.8 Modelo eventos.

3.4.8.1 *Proceso creación.*

Modelo de Caso de Uso #16.



Escenario de Caso de Uso #16.

Tabla 19. Escenario de Caso de Uso, Proceso Creación de Eventos

ACTORES	Súper Administrador Administrador
PRECONDICIONES	Predefinir fechas del evento.
POSCONDICIONES	Asignación de Eventos.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: 1. El usuario autorizado o administrador realiza la creación del evento o reunión mediante su planificación.

continúa...

Tabla 19. Escenario de Caso de Uso (continuación...)

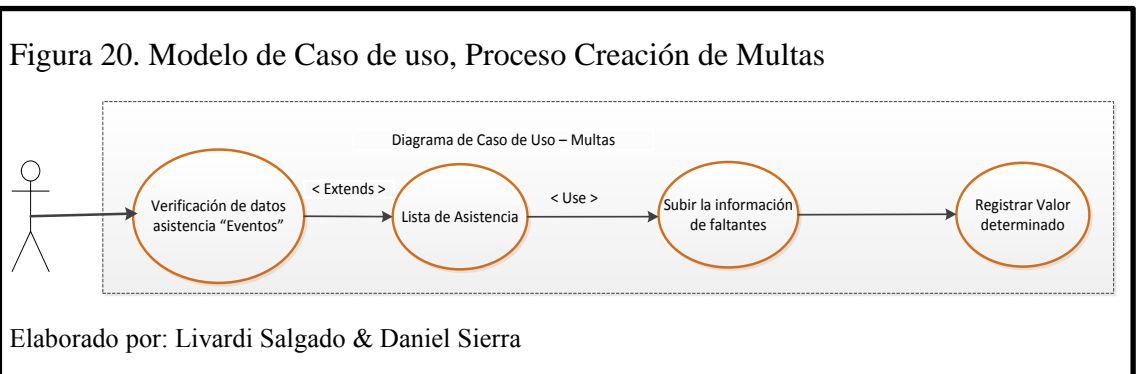
	2. Una vez organizado se crea el evento y genera una lista de asistencia de las personas de la comunidad.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos ingresados son incorrectos la información no es almacenada.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.9 Modelo multas.

3.4.9.1 Proceso creación.

Modelo de Caso de Uso #17.



Escenario de Caso de Uso #18.

Tabla 20. Escenario de Caso de Uso, Proceso Creación de Perfil

ACTORES	Administrador Usuario
OBJETIVO	Consultar y generar la factura de un cliente tomando en cuenta los parámetros para generar la misma.
PRECONDICIONES	Haber ingresado al sistema.
POSCONDICIONES	El usuario y/o administrador realizará la validación de información para emitir la factura correspondiente para el cliente.
FLUJO BÁSICO	<p>El flujo básico debe ser de la siguiente forma:</p> <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la facturación mediante la búsqueda del código de medidor en el sistema proporcionado por el cliente2. Una vez encontrada la factura la misma es procesada e impresa hacia el cliente final.
FLUJO ALTERNATIVO	<p>El flujo básico debe ser de la siguiente forma:</p> <p>2.1 Si los datos del código de medidor es incorrecto la facturación no es procesada.</p>

Elaborado por: Livardi Salgado & Daniel Sierra

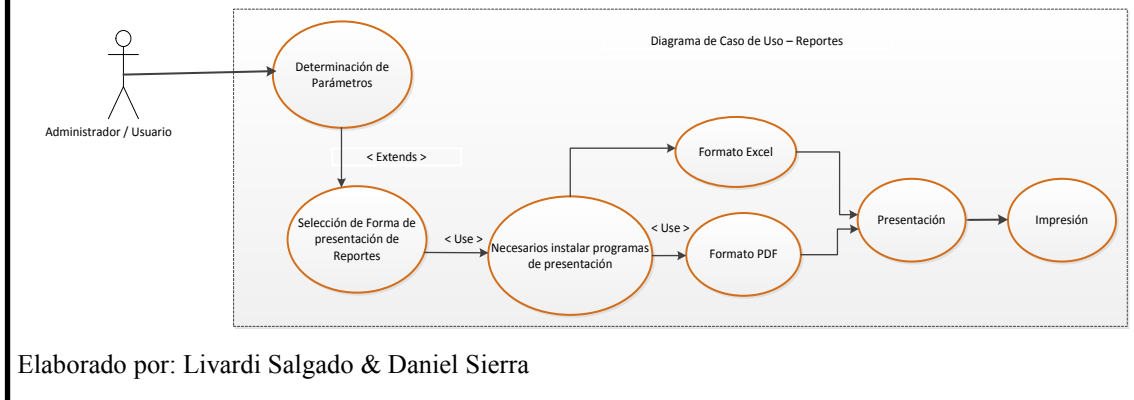
3.4.10 Modelo reportes.

3.4.10.1 Proceso generación.

En este módulo el usuario tendrá la facilidad de sacar un reporte dependiendo de las necesidades o requerimientos de los clientes o usuarios finales. El usuario ingresa los parámetros de consulta, los mismos que son enviados a la base de datos para obtener un histórico de lo requerido.

Modelo de Caso de Uso #19.

Figura 21. Modelo de Caso de Uso – Generación de Reportes



Elaborado por: Livardi Salgado & Daniel Sierra

Escenario de Caso de uso # 19.

Tabla 21. Escenario de Caso de Uso, Proceso Creación de Perfil

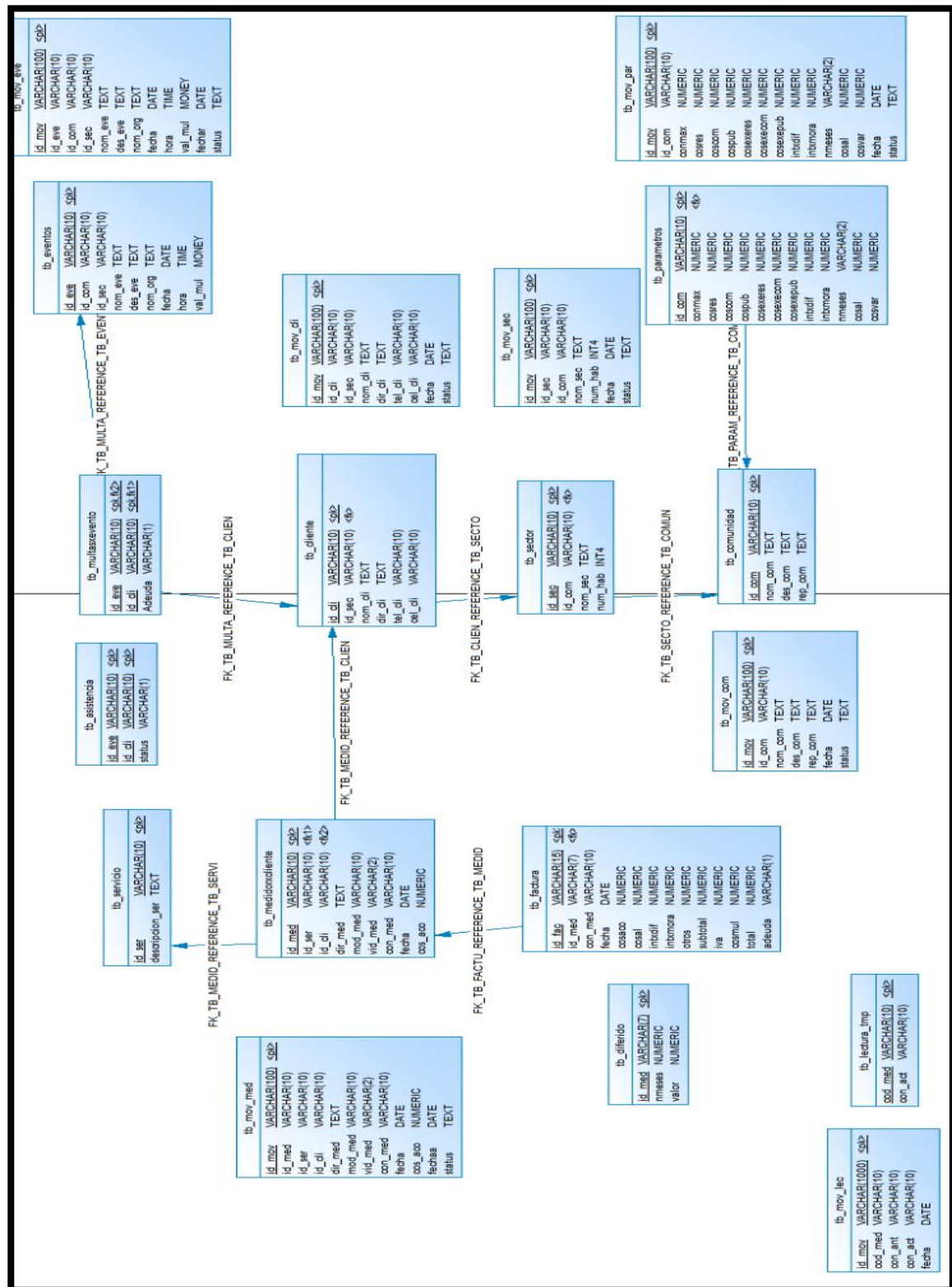
ACTORES	Administrador Usuario
PRECONDICIONES	Haber ingresado al sistema.
POSCONDICIONES	El usuario y/o administrador realizará la validación de información para emitir la factura correspondiente para el cliente.
FLUJO BÁSICO	El flujo básico debe ser de la siguiente forma: <ol style="list-style-type: none">1. El usuario autorizado o administrador realiza la búsqueda de la información deseada en cada módulo de acuerdo a su necesidad.2. Cada módulo tiene sus parámetros de búsqueda para obtener la información solicitada en formatos de impresión de tipo Excel o PDF.
FLUJO ALTERNATIVO	El flujo básico debe ser de la siguiente forma: 2.1 Si los datos de búsqueda no coinciden, el reporte no será generado.

Elaborado por: Livardi Salgado & Daniel Sierra

3.5 Diagrama de base de datos

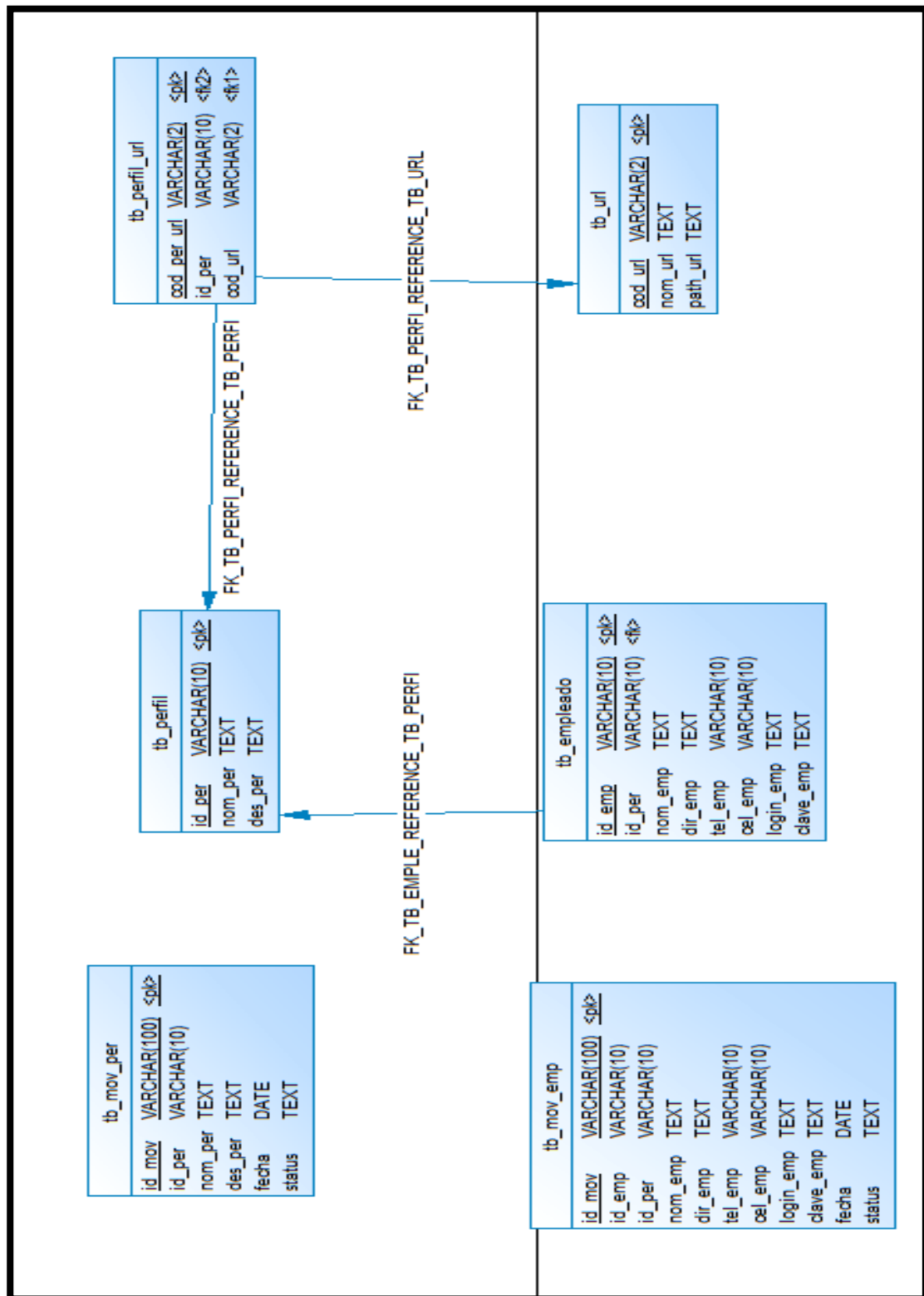
En los siguientes diagramas menciona las relaciones principales entre tablas que permiten la comunicación entre ellas, adicional la base de datos completa.

Figura 22. Diagrama físico de la base de datos relacionada



Elaborado por: Livardi Salgado & Daniel Sierra

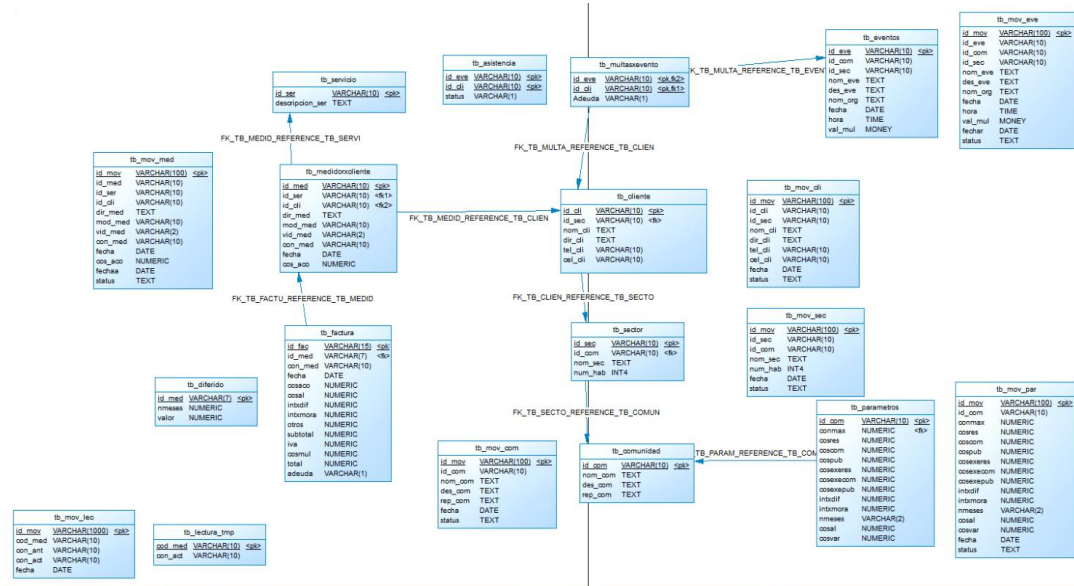
Figura 23. Diagrama Integro de base de datos



Elaborado por: Livardi Salgado & Daniel Sierra

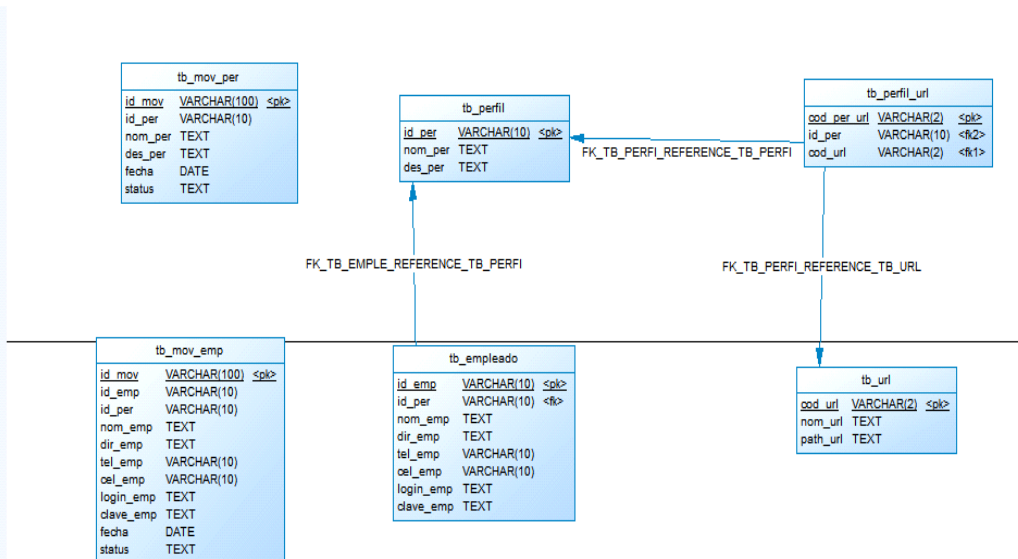
3.5.1 Diagrama entidad relación.

Figura 24. Diagrama físico de la base de datos



Elaborado por: Livardi Salgado & Daniel Sierra

Figura 25. Diagrama físico de la base de datos relacionada



Elaborado por: Livardi Salgado & Daniel Sierra

CAPÍTULO 4

IMPLEMENTACIÓN

Durante este capítulo, se muestra la codificación de las diferentes clases que permiten construir los módulos previamente diseñados.

4.1 Codificación

Describe los fragmentos más importantes del desarrollo del sistema de facturación.

4.1.1 Clase: conexión.

Permite establecer una conexión entre la aplicación con la base de datos mediante la invocación de código fuente.

Figura 26. Código Fuente – Clase Conexión

```
public Conexion() {  
  
    this.driver ="org.postgresql.Driver";  
    this.user="postgres";  
    this.pwd="ppiups2014";  
    this.cadena="jdbc:postgresql://localhost:5432/PPI_Facturacion";  
    this.con=this.crearConexion();  
  
}  
  
Connection crearConexion()  
{  
    try {  
        Class.forName("org.postgresql.Driver");  
    }  
    catch (ClassNotFoundException e) {  
  
    }  
  
    try  
    {  
        Class.forName(getDriver()).newInstance();  
        Connection con=DriverManager.getConnection(getCadena(),getUser(),getPwd());  
        return con;  
    }  
    catch(Exception ee)
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se puede identificar los métodos de conexión hacia la base de datos por medio de un conector JDBC hacia Postgresql.

Figura 27. Código Fuente – Clase Conexión JDBC

```
    {
        System.out.println("Error: " + ee.getMessage());
        return null;
    }
}

public String Ejecutar(String sql)
{
    String error="";
    try
    {
        St=getConexion().createStatement();
        St.execute(sql);
        error="Registro";
    }
    catch(Exception ex)
    {
        error = "Error";
    }
    return(error);
}

public ResultSet Consulta(String sql)
{
    ResultSet reg=null;
    try
    {
        St=getConexion().createStatement();
        reg=St.executeQuery(sql);
    }
    catch(Exception ee)
    {
        System.out.println(ee.getMessage());
    }
    return(reg);
}
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se puede observar la creación de sentencias de conexión y resultados de la búsqueda en la base de datos.

4.1.2 Clase: parámetros.

Figura 28. Código Fuente – Clase Parámetros

```
/*METODO DE REGISTRO DE PARAMETROS*/
public void registro(Datosparametros dp){

    try {

        /*INGRESO DE DATOS A LA TABLA PARAMETROS*/
        String sentencia="Insert into tb_parametros values('"+dp.getIdcom()+"','"+dp.getIdcomax()+"','"+dp.getIdcosres()+"','"+dp.getIdcoscom()+"','"+dp.getIdcospub()+"','"+dp.getIdcosexeres()+"','"+dp.getIdcosresregl()+"','"+dp.getIdcosresregl2()+"')";
        resregl=clsCon.Ejecutar(sentencia);

        /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIENTOS*/
        rs=clsCon.Consulta("select '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_par");
        rs.next();
        String mov=rs.getString(1);

        /*INGRESO DE DATOS A LA TABLA DE MOVIENTOS*/
        String sentencia2="Insert into tb_mov_par values('"+mov+"','"+dp.getIdcom()+"','"+dp.getIdcomax()+"','"+dp.getIdcosres()+"','"+dp.getIdcoscom()+"','"+dp.getIdcospub()+"','"+dp.getIdcosexeres()+"','"+dp.getIdcosresregl()+"','"+dp.getIdcosresregl2()+"')";
        resregl2=clsCon.Ejecutar(sentencia2);

    } catch (Exception e1) {
        e1.getMessage();
    }
}

/*METODOS DE CONFIGURACION REGISTROS*/
public String getResregl(){
    return this.resregl;
}

public String getResregl2(){
    return this.resregl2;
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se verifica la recepción de parámetros necesarios para describir las necesidades de cada comunidad.

Figura 29. Código Fuente – Clase Parámetros – Funciones Varias

```
try {

    /*ACTUALIZACION DE DATOS A LA TABLA PARAMETROS*/
    String sentencia="Update tb_parametros SET comax='"+dp.getIdcomax()+"',cosres='"+dp.getIdcosres()+"',coscom='"+dp.getIdcoscom()+"',cospub='"+dp.getIdcospub()+"',cosexeres='"+dp.getIdcosexeres()+"' where id_com='"+dp.getIdcom()+"'";
    respectul=clsCon.Ejecutar(sentencia);

    /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIENTOS*/
    rs=clsCon.Consulta("select '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_par");
    rs.next();
    String mov=rs.getString(1);

    /*INGRESO DE DATOS A LA TABLA DE MOVIENTOS*/
    String sentencia2="Insert into tb_mov_par values('"+mov+"','"+dp.getIdcom()+"','"+dp.getIdcomax()+"','"+dp.getIdcosres()+"','"+dp.getIdcoscom()+"','"+dp.getIdcospub()+"','"+dp.getIdcosexeres()+"','"+dp.getIdcosresregl()+"','"+dp.getIdcosresregl2()+"')";
    resregl2=clsCon.Ejecutar(sentencia2);

} catch (Exception e1) {
    e1.getMessage();
}

/*METODOS DE CONFIGURACION DE ACTUALIZACION*/
public String getRespectul(){
    return this.respectul;
}

public String getRespectul2(){
    return this.respectul2;
}

public void eliminar(Datosparametros dp){
    /*CONSULTAS E INGRESOS A BASE DE DATOS*/
    try {

        /*ELIMINACION DE DATOS A LA TABLA PARAMETROS*/
        String sentencia="Delete from tb_parametros where id_com='"+dp.getIdcom()+"'";
        resregl=clsCon.Ejecutar(sentencia);

        /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIENTOS*/
        rs=clsCon.Consulta("select '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_par");
        rs.next();
        String mov=rs.getString(1);

        /*INGRESO DE DATOS A LA TABLA DE MOVIENTOS*/
        String sentencia2="Insert into tb_mov_par values('"+mov+"','"+dp.getIdcom()+"','"+dp.getIdcomax()+"','"+dp.getIdcosres()+"','"+dp.getIdcoscom()+"','"+dp.getIdcospub()+"','"+dp.getIdcosexeres()+"','"+dp.getIdcosresregl()+"','"+dp.getIdcosresregl2()+"')";
        resregl2=clsCon.Ejecutar(sentencia2);

    } catch (Exception e1) {
        e1.getMessage();
    }
}

/*METODOS DE CONFIGURACION DE ELIMINACION*/
public String getResregl1(){
    return this.resregl1;
}

public String getResregl2(){
    return this.resregl2;
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se describe las sentencias y código fuente que será enviado a la base de datos para la recolección de información.

4.1.3 Clase administradores.

Figura 30.: Código Fuente – Clase Administrador

```
public void registro(Datosadmin da){
    /*CONSULTA E INGRESO A BASE DE DATOS*/
    try {
        /*INGRESO DE DATOS A LA TABLA EMPLEADOS*/
        String sentencia1="Insert into tb_empleado values('"+da.getCedula()+"','"+da.getPerfil()+"','"+da.getNombre()+"','"+da.getDireccion()+"','"+da.getTelefono()+"','"+da.getCelular()+"','"+da.getIngir
        respreg1=clsCon.Ejecutar(sentencia1);

        /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIZIENTOS*/
        rs=clsCon.Consulta("select '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_emp");
        rs.next();
        String mov=rs.getString(1);

        /*INGRESO DE DATOS A LA TABLA DE MOVIZIENTOS*/
        String sentencia2="Insert into tb_mov_emp values('"+mov+"','"+da.getCedula()+"','"+da.getPerfil()+"','"+da.getNombre()+"','"+da.getDireccion()+"','"+da.getTelefono()+"','"+da.getCelular()+"','"+da.getIngir
        respreg2=clsCon.Ejecutar(sentencia2);

    } catch (Exception ex) {
        ex.getMessage();
    }
}

/*METODOS DE CONFIRMACION REGISTRO*/
public String getRespreg1(){
    return this.respreg1;
}

public String getRespreg2(){
    return this.respreg2;
}

/*-----*/
public void actualizar(Datosadmin da){
    /*CONSULTA E INGRESO A BASE DE DATOS*/
    try {
        /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIZIENTOS*/
        rs=clsCon.Consulta("select '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_emp");
        rs.next();
        String mov=rs.getString(1);
    }
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se puede observar la creación de usuarios administradores para la gestión de usuarios.

4.1.4 Clase: facturas.

Figura 30. Código Fuente – Clase Factura

```
/*METODO PARA PROCESAR LA FACTURA MENSUALMENTE*/
public void procesarFactura(Datosfactura dfac){
    try {
        rst=clsCon.Consulta("select * from tb_lectura_tmp");
        while (rst.next())
        {
            cosmul=0;
            valdif=0;
            idmed=rst.getString(1);
            contadora=rst.getString(2);

            /*OBTENCION DEL SUMINISTRO DEL MEDIDOR*/
            rsmed=clsCon.Consulta("SELECT * FROM tb_medidorxcliente where id_med='"+idmed+"'");
            while (rsmed.next())
            {
                idser=rsmed.getString(2);
                idcli=rsmed.getString(3);
                contadori=Integer.parseInt(rsmed.getString(7));
                cosaco=Double.parseDouble(rsmed.getString(9));
            }

            /*CALCULO DE CONSUMO ACTUAL Y ANTERIOR*/
            consumo=Integer.parseInt(contadora)-contadori;

            /*OBTENCION ID DEL SECTOR*/
            rssec=clsCon.Consulta("SELECT * FROM tb_cliente where id_cli='"+idcli+"'");
            while (rssec.next())
            {
                idsec=rssec.getString(2);
            }

            /*OBTENCION DE MULTAS DEL CLIENTE*/
            rsmul=clsCon.Consulta("SELECT * FROM tb_multasevento where id_cli='"+idcli+"' AND adeuda='S'");
            while (rsmul.next())
            {
                ideve=rsmul.getString(1);
                rsvalmul=clsCon.Consulta("SELECT * FROM tb_eventos where id_eve='"+ideve+"'");
                while (rsvalmul.next())
                {
                    cosmul=cosmul+Double.parseDouble(rsvalmul.getString(9));
                }
            }

            /*OBTENCION DEL ID DE LA COMUNIDAD*/
            rscow=clsCon.Consulta("SELECT * FROM tb_sector where id_sec='"+idsec+"'");
            while (rscow.next())
            {
                idcom=rscow.getString(2);
            }
        }
    }
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se puede identificar la obtención, procesamiento y cálculo de los valores de la factura para las comunidades cumpliendo los requerimientos de los parámetros realizados.

Figura 31. Código Fuente – Clase Factura – Función Métodos varios

```
/*APLICACION DE INTERESES POR DIFERIR*/
rsint=clsCon.Consulta("SELECT * FROM tb_diferido where id_med='"+idmed+"'");
if(rsint.next()==false){
    intdif=0;
}
else
{
    valdif=Double.parseDouble(rsint.getString(3));
}

/*APLICACION DE INTERESES POR MORA*/
int i=0;
rsintm=clsCon.Consulta("SELECT * FROM tb_factura where id_med='"+idmed+"' AND status='PENDIENTE'");
while (rsintm.next()){
    i++;
}
if(i>0){
    //pw.print("Adeuda pagos");
}
else
{
    //pw.print("Al día en pagos");
    intxmora=0;
}
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se puede observar el cálculo de interés a diferir y el cálculo de interés por mora del usuario.

Figura 32. Código Fuente – Clase Factura – Función Cálculo

```
/*MÉTODOS PARA PAGO DE FACTURA PENDIENTE*/
public void pagarFactura(Date factura dfac){
    try {
        rsfac=clsCon.Consulta("SELECT * FROM tb_diferido where id_med='"+dfac.getIdMed()+"'");
        while (rsfac.next()){
            mesres=Integer.parseInt(rsfac.getString(2));
            if(dfac.getMeses())<=1 && mesres=1){
                String sentencia="update tb_factura SET status='"+dfac.getStatus()+"' where id_fac='"+dfac.getIdFac()+"'";
                respago=clsCon.Ejecutar(sentencia);
                String sentencia2="update tb_diferido SET mesres='"+(mesres-1)+"' where id_med='"+dfac.getIdMed()+"'";
                respago=clsCon.Ejecutar(sentencia2);
                String sentencia3="update tb_multasevento SET adeuda='N' where id_cil='"+dfac.getIdCil()+"'";
                respago=clsCon.Ejecutar(sentencia3);
            }
            else
            {
                if(dfac.getMeses())<=1 && mesres=2){
                    String sentencia="update tb_factura SET status='"+dfac.getStatus()+"' where id_fac='"+dfac.getIdFac()+"'";
                    respago=clsCon.Ejecutar(sentencia);
                    String sentencia2="delete from tb_diferido where id_med='"+dfac.getIdMed()+"'";
                    respago=clsCon.Ejecutar(sentencia2);
                    String sentencia3="update tb_multasevento SET adeuda='N' where id_cil='"+dfac.getIdCil()+"'";
                    respago=clsCon.Ejecutar(sentencia3);
                }
                else
                {
                    if(dfac.getMeses())<=1 && mesres=0){
                        String sentencia="insert into tb_diferido values ('"+dfac.getIdMed()+"','"+dfac.getMeses()+"','"+dfac.getIdFac()+"','"+dfac.getIdCil()+"')";
                        respago=clsCon.Ejecutar(sentencia);
                        String sentencia2="update tb_factura SET tapago='"+dfac.getIdMed()+"',status='"+dfac.getStatus()+"' where id_fac='"+dfac.getIdFac()+"'";
                        respago=clsCon.Ejecutar(sentencia2);
                        String sentencia3="update tb_multasevento SET adeuda='N' where id_cil='"+dfac.getIdCil()+"'";
                        respago=clsCon.Ejecutar(sentencia3);
                    }
                    else
                    {
                        if(dfac.getMeses())<=1 && mesres=0){
                            String sentencia="update tb_factura SET tapago='"+dfac.getIdMed()+"',status='"+dfac.getStatus()+"' where id_fac='"+dfac.getIdFac()+"'";
                            respago=clsCon.Ejecutar(sentencia);
                            String sentencia2="update tb_multasevento SET adeuda='N' where id_cil='"+dfac.getIdCil()+"'";
                            respago=clsCon.Ejecutar(sentencia2);
                            respago="fin";
                        }
                    }
                }
            }
        }
    } catch (Exception ex) {
        ex.getMessage();
    }
}

/*MÉTODOS DE CONFIRMACION REGISTRO*/
public String getRespago(){
    return this.respago;
}
}
```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se identifica el cálculo de pago de una factura mensual, el cual determina los valores reales para el usuario final.

4.1.5 Clase: medidores.

Figura 33. Código Fuente – Clase Medidor

```

    "/METHODO DE REGISTRO DE MOVIMIENTOS"/
    public void registro(Gatosmedidor dm){

        /*CONSULTAS E INGRESOS A BASE DE DATOS*/
        try {
            "/INGRESO DE DATOS EN LA TABLA MOVIMIENTOSCLIENTE"/
            String sentencia="Insert into tb_movimiento values('"+dm.getIdmed()+"','"+dm.getIdrefr()+"','"+dm.getIdcli()+"','"+dm.getIdreed()+"','"+dm.getIdvmed()+"')";
            rs=conexion.getConnection().executeUpdate(sentencia);

            "/EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIMIENTOS"/
            rs=conexion.getConnection().executeQuery("select ' '||(max(cast(substr(id_mov,1) as int)+1)) from tb_mov_mnd");
            rs.next();
            String nuevo=rs.getString(1);

            "/INGRESO DE DATOS A LA TABLA DE MOVIMIENTOS"/
            String sentencia2="Insert into tb_mov_mnd values('"+nuevo+"','"+dm.getIdmed()+"','"+dm.getIdrefr()+"','"+dm.getIdcli()+"','"+dm.getIdreed()+"','"+dm.getIdvmed()+"')";
            rs2=conexion.getConnection().executeUpdate(sentencia2);

        } catch (Exception e1) {
            e1.getMessage();
        }

    }

    /*METODOS DE CONFIRMACION REGISTRO*/
    public String getrespreg1(){
        return this._respreg1;
    }

    public String getrespreg2(){
        return this._respreg2;
    }
}

```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se identifica el ingreso y registro de medidores que es asociada a un cliente previamente creado.

4.1.6 Clase: lecturas.

Figura 34. Código Fuente – Clase Lectura

[illegible]

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura se muestra la información de ingreso de registros de medición a la base de datos que permitirá obtener el cálculo de la última lectura con la nueva información, de esta forma el usuario tiene el registro real automatizado de ingreso de información.

4.1.7 Clase: eventos.

Figura 35. Código Fuente – Clase Eventos

```

/*METODO DE REGISTRO DE EVENTOS*/
public void registro(Datosevento de){
    /*CONSULTAS E INGRESOS A BASE DE DATOS*/
    try {
        /*DECLARACION DE REGISTROS EN LA TABLA ASISTENCIA*/
        String sentencia="Delete from tb_asistencia";
        respreg=clsCon.Ejecutar(sentencia);

        /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE EVENTOS*/
        rs=clsCon.Consulta("select 'Eve-'||(max(cast(substr(id_eve,5) as int)+1)) from tb_eventos");
        while (rs.next()){
            ideve=rs.getString(1);

            /*REGISTRO DEL EVENTO*/
            String sentencia="Insert into tb_eventos values('"+id eve+"','"+de.getIdcom()+"','"+de.getIdsec()+"','"+de.getIdhome()+"','"+de.getIdseve()+"','"+de.getIdomorg()+"','"+
            respreg=clsCon.Ejecutar(sentencia);

            /*EXTRACCION DEL LA INFORMACION DE CLIENTES EN EL SECTOR*/
            rs=clsCon.Consulta("select * from tb_cliente where id_sec='"+de.getIdsec()+"");
            while (rs.next()){
                String idcli=rs.getString(1);
                String nomcli=rs.getString(2);
                String dircli=rs.getString(4);

                /*REGISTRO DE TODOS LOS USUARIOS DENTRO DEL SECTOR*/
                String sentencia="Insert into tb_asistencia values('"+id eve+"','"+id cli+"','"+nomcli+"','"+dircli+"');
                respreg=clsCon.Ejecutar(sentencia);
            }

            /*EXTRACCION DEL ULTIMO REGISTRO DE LA TABLA DE MOVIMIENTOS*/
            rs=clsCon.Consulta("select 'M'+(cast(substr(id_mov,1) as int)+1)) from tb_mov_eve");
            rs.next();
            String mov=rs.getString(1);

            /*INGRESO DE DATOS A LA TABLA DE MOVIMIENTOS*/
            String sentencia="Insert into tb_mov_eve values('"+mov+"','"+id eve+"','"+de.getIdcom()+"','"+de.getIdsec()+"','"+de.getIdhome()+"','"+de.getIdseve()+"','"+de.getIdomorg()+"');
            respreg=clsCon.Ejecutar(sentencia);
        }
    } catch (Exception e1) {
        e1.getMessage();
    }
}

```

Elaborado por: Livardi Salgado & Daniel Sierra

En la figura, se muestra la configuración para guardar los eventos de la comunidad (Mingas o juntas de trabajo) asignando una fecha y hora del mismo, el que permite obtener la información de las personas que deben asistir al evento.

4.1.8 Clase: asistencia.

Figura 36. Código Fuente – Clase Asistencia

```

# Import metodos.Datosasistencia;
public class Asistencia {
    /*DECLARACION DE VARIABLES*/
    ResultSet rs;
    Conexion clsCon=new Conexion();
    String respreg;

    public void registro(Datosasistencia da){
        /*CONSULTAS E INGRESOS A BASE DE DATOS*/
        try {
            rs=clsCon.Consulta("select * from tb_eventos, tb_cliente where id_eve='EVE-0' AND tb_cliente.id_sec=tb_eventos.id_sec AND tb_eventos.id_com='"+da.getIdcom()+"' AND ");
            if(rs.next()==false){
                /*INGRESO DE DATOS A LA TABLA MULTASXEVENTO*/
                String sentencia="Insert into tb_multasxevento values('"+da.getId eve()+"','"+da.getId cli()+"','"+da.getId seve()+"');
                respreg=clsCon.Ejecutar(sentencia);
            }
            else{
                respreg="Error";
            }
        } catch (Exception e1) {
            e1.getMessage();
        }
    }

    /*METODOS DE CONFIRMACION REGISTRO*/
    public String getRespregi(){
        return this.respreg;
    }
}

```

Elaborado por: Livardi Salgado & Daniel Sierra

La figura, permite registrar la inasistencia de las personas que no acudieron al evento programado, es con el fin de registrarlo y aplicar la multa respectiva.

4.1.9 Clase: perfiles.

Figura 37. Código Fuente – Clase Perfiles

```

/*METODO DE REGISTRO DE COMUNIDADES*/
public void registro(Datosperfil dp){

/*CONSULTAS E INGRESOS A BASE DE DATOS*/
try {
    rsc=clsCon.Consulta("select * from tb_perfil where nom_per="+dp.getNomper()+"");
    if(rsc.next()==false){
        rs=clsCon.Consulta("select ''||(max(cast(substr(id_per,1) as int)+1)) from tb_perfil");
        rs.next();
        String idper=rs.getString(1);
        String sentencia="Insert into tb_perfil values('"+idper+"','"+dp.getNomper()+"','"+dp.getDesper()+"')";
        respregi=clsCon.Ejecutar(sentencia);

        if(dp.getAdministrador()!=null){
            rs1=clsCon.Consulta("select ''||(max(cast(substr(cod_per_url,1) as int)+1)) from tb_perfil_url");
            rs1.next();
            String idcodur1=rs1.getString(1);
            String sentenciao="Insert into tb_perfil_url values('"+idcodur1+"','"+idper+"','"+1+"')";
            String respuestaa=clsCon.Ejecutar(sentenciao);}

        if(dp.getPerfil()!=null){
            rs2=clsCon.Consulta("select ''||(max(cast(substr(cod_per_url,1) as int)+1)) from tb_perfil_url");
            rs2.next();
            String idcodur2=rs2.getString(1);
            String sentenciap="Insert into tb_perfil_url values('"+idcodur2+"','"+idper+"','"+2+"')";
            String respuestap=clsCon.Ejecutar(sentenciap);}

        if(dp.getUsuario()!=null){
            rs3=clsCon.Consulta("select ''||(max(cast(substr(cod_per_url,1) as int)+1)) from tb_perfil_url");
            rs3.next();
            String idcodur3=rs3.getString(1);
            String sentenciau="Insert into tb_perfil_url values('"+idcodur3+"','"+idper+"','"+3+"')";
            String respuestau=clsCon.Ejecutar(sentenciau);}

        if(dp.getMedidor()!=null){
            rs4=clsCon.Consulta("select ''||(max(cast(substr(cod_per_url,1) as int)+1)) from tb_perfil_url");
            rs4.next();
            String idcodur4=rs4.getString(1);
            String sentenciam="Insert into tb_perfil_url values('"+idcodur4+"','"+idper+"','"+4+"')";
            String respuestam=clsCon.Ejecutar(sentenciam);}
    }
}

```

Elaborado por: Livardi Salgado & Daniel Sierra

La figura permite la asignación de perfiles para determinar las labores y funciones a realizar por los usuarios creados.

CAPÍTULO 5

PRUEBAS

En este capítulo se procede a realizar las diferentes pruebas de funcionalidad y adaptación con el usuario final detectando posibles errores, adicional a ello se realiza pruebas de estrés analizando el rendimiento y carga de la aplicación en un equipo con el fin de determinar las características óptimas para su funcionamiento.

5.1 Pruebas funcionales

Se realiza una prueba funcional o un test al software creado que permite comparar los requerimientos de los usuarios en base a las funciones que el sistema permite registrar

5.1.1 Creación y asignación.

Tabla 22. Prueba Funcional – Proceso Autenticación

CASO AUTENTICACIÓN DE USUARIO						
Número de prueba	Descripción del caso de Prueba	Pre-Requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_LO GI	Ingreso al sistema con Usuario Administrador	Usuario registrado en el sistema	Ingreso Exitoso	Ingreso Correcto	Procesado	
PP_02_LO GIN	Ingreso al sistema con Usuario registrado	Usuario registrado en el sistema	Ingreso Exitoso	Ingreso Correcto	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 23. Prueba Funcional – Proceso Generación de Reportes

CASO DE GENERACIÓN DE REPORTES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_gr	Generación de reportes	Usuario registrado e ingresado al sistema con perfil de consultas	Reporte Generado	Reporte Generado	Procesado	Ingreso de parámetros de consulta de fechas inferiores a la actual.

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 24. Prueba Funcional – Proceso Creación de Usuarios

CASO CREACIÓN DE USUARIOS						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_cus uario	Creación de usuario Administrador	Datos del usuario	Creación Satisfactoria	Creación realizada	Procesado	
PP_02_cus uario	Creación de usuarios (Cajeros, administrativos)	Datos del usuario	Creación Satisfactoria	Creación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 25. Prueba Funcional – Proceso Asignación de Perfiles

CASO ASIGNACIÓN DE PERFILES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_aprfiles	Asignación de perfil	Identificar sus funciones de acuerdo a los perfiles establecidos	Asignación correcta	Asignación realizada	Procesado	
PP_02_aprfiles	Asignación de dos o más perfiles	Identificar sus funciones de acuerdo a los perfiles establecidos	Asignación correcta	Asignación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 26. Prueba Funcional – Proceso Creación de Clientes

CASO CREACIÓN DE CLIENTES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_cclientes	Creación de clientes	Datos del cliente	Creación Satisfactoria	Creación realizada	Procesado	
PP_02_cclientes	Creación de clientes	Datos del cliente	Creación Satisfactoria	Creación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 27. Prueba Funcional – Proceso Medidores

CASO CREACIÓN DE MEDIDORES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_cmedidor	Asignación de un medidor a un usuario registrado	Creación del usuario	Registrar medidor	Medidor registrado	Procesado	
PP_02_cmedidor	Asignación de un medidor a un usuario registrado	Datos del usuario	Registrar medidor	Medidor registrado	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 28. Prueba Funcional – Proceso Creación de Eventos

CASO CREACIÓN DE EVENTOS						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_ceventos	Creación de un evento	Determinación de fechas	Evento Creado	Evento registrado	Procesado	
PP_02_ceventos	Asignación de un evento adicional	Determinar fechas con disponibilidad	Evento Creado	Evento registrado	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 29. Prueba Funcional – Proceso Registro de Multas

CASO REGISTRO DE MULTAS						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_rm ltas	Asignación de multas	Tabla de asistencias de evento realizado	Multa asignada	Multa registrada	Procesado	
PP_02_rm ltas	Asignación de multas	Tabla de eventos	Multa asignada	Evento registrado	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

5.1.2 Actualización.

Tabla 30. Prueba Funcional – Proceso Actualización de Usuarios

CASO ACTUALIZACIÓN DE USUARIOS						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_aus uario	Actualización de usuario Administrador	Búsqueda de usuario anterior	Actualización Satisfactoria	Creación realizada	Procesado	
PP_02_aus uario	Actualización de usuarios (Cajeros, administrativos)	Datos de búsqueda	Actualización Satisfactoria	Creación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 31. Prueba Funcional – Proceso Actualización de Clientes

CASO ACTUALIZACIÓN DE CLIENTES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_acientes	Actualización de clientes	Datos del cliente(cedula)	Actualización Satisfactoria	Actualización realizada	Procesado	
PP_02_acientes	Actualización de clientes	Datos del cliente(cedula)	Actualización Satisfactoria	Actualización realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 32. Prueba Funcional – Proceso Actualización de medidores

CASO ACTUALIZACIÓN DE MEDIDORES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_amedidor	A Actualización de un medidor a un usuario registrado	Datos del usuario y numero de medidor	Actualización medidor	Medidor Actualizado	Procesado	
PP_02_amedidor	Actualización de un medidor a un usuario registrado	Datos del usuario y numero de medidor	Actualización medidor	Medidor r Actualizado	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

5.1.3 Eliminación de registros.

Tabla 33. Prueba Funcional – Proceso Eliminación de Usuarios

CASO ELIMINACIÓN DE USUARIOS						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_usuario	Eliminación de usuario Administrador	Búsqueda de usuario registrado	Eliminación Satisfactoria	Eliminación realizada	Pendiente	Contiene perfiles ligados
PP_02_usuario	Actualización de usuarios (Cajeros, administrativos)	Datos de búsqueda	Eliminación Satisfactoria	Eliminación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 34. Prueba Funcional – Proceso Eliminación de clientes

CASO ELIMINACIÓN DE CLIENTES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_clientes	Eliminación de cliente Administrador	Búsqueda de cliente registrado	Eliminación Satisfactoria	Eliminación realizada	Pendiente	
PP_02_clientes	Eliminación	Datos de búsqueda	Eliminación Satisfactoria	Eliminación realizada	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

Tabla 35. Prueba Funcional – Proceso Eliminación de Medidores

CASO ELIMINACIÓN DE MEDIDORES						
Número de prueba	Descripción del caso de prueba	Pre-requisito	Resultado esperado	Resultado obtenido	Estado	Observación
PP_01_eliminator	Eliminación de un medidor a un usuario	Datos del usuario y numero de	Eliminación medidor	Medidor Eliminado	Procesado	
PP_02_eliminator	Eliminación de un medidor a un usuario registrado	Datos del usuario y numero de medidor	Eliminación medidor	Medidor eliminado	Procesado	

Elaborado por: Livardi Salgado & Daniel Sierra

5.2 Pruebas de conexión

Se realizó las pruebas de conexión hacia el aplicativo por medio de un servidor, una conexión de red cableada conectada a través de un Switch de comunicaciones y las maquinas con una conexión de red con el puerto 8080 y 8082 abierto, durante las pruebas de conexión se identificó un enlace estable mediante testeo de Ping.

Para el testeo realizado se ocuparon los siguientes equipos de comunicaciones

Tabla 36. Descripción de Equipos Servidor y Red

INTERFACE	WEB
IP ALOJAMIENTO	http://localhost:8082/proyectopesillo/Login.jsp
DOMINIO	Personal
SERVIDOR	APACHE
NUMERO DE EQUIPOS	5
SW DE RED	CISCO 3950
CONEXIÓN	RED CABLEADA
EQUIPO	HP 8300 SMALL FORM FACTOR

PROCESADOR	i7
DISCO DURO	1 TB
MEMORIA	8 GB RAM
RED	Interface de Red

Elaborado por: Livardi Salgado & Daniel Sierra

Figura 38. Descripción de Equipos Clientes

Computer	User	Operating system	RAM (MB)	CPU (MHz)
DTV-UIO-ASIS-02		Microsoft Windows 7 Professional	3072	2200
DTV-UIO-CEXP-01		Microsoft Windows 7 Professional	4096	3401
DTV-UIO-CEXP-03		Microsoft Windows 7 Professional	4096	2501
DTV-UIO-CEXP-04		Microsoft Windows 7 Professional	4096	3401
DTV-UIO-CEXP-05		Microsoft Windows 7 Professional	4096	3333
DTV-UIO-CEXP-08		Microsoft Windows 7 Professional	6144	3101
DTV-UIO-CEXP-10		Microsoft Windows 7 Professional	3072	2401
DTV-UIO-CEXP-11		Microsoft Windows 7 Professional	3072	2100
DTV-UIO-CEXP-12		Microsoft Windows 7 Professional	2766	2200
DTV-UIO-CEXP-14		Microsoft Windows 7 Professional	4096	3101
DTV-UIO-CEXP-16		Microsoft Windows 7 Professional	4096	3401

Elaborado por: Livardi Salgado & Daniel Sierra

Figura 39. Información relacionada de Equipos

Name:	DTV-UIO-CEXP-01	Domain:	DTVPAN.COM
IP address:	172.22.150.151 WOL	Swap:	7940
OS Name:	Microsoft Windows 7 Professional	OS Version:	6.1.7601
Service pack:	Service Pack 1	Windows user:	admin
Windows licence:	S5041-008-1508806-86601	Windows key:	MAK or VLK key unavailable
User agent:	OCS-NG_WINDOWS_AGENT_v2.1.0.3	Memory:	4096
Last inventory:	05/12/2014 11:57:49	Last contact:	05/12/2014 11:57:49
Description:	Andres Paguay	Uuid:	B3B7B0B0-EEA5-11E1-AED6-24BE05028DAA
Architecture:	x86 64 bit	Export:	XML

Elaborado por: Livardi Salgado & Daniel Sierra

Figura 40. Características del Equipo

Manufacturer	Type	Serial number	Frequency	Cores number	L2 cache size	Architecture	Data width	Current address width	Logical CPUS	Voltage	Current speed	Socket type
GenuineIntel	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz	CPU Enabled	3401	4	1024	x86_64	64	64	8		3401	SOCKET 0

MEMORY

4 Result(s) (Download)

Caption	Description	Capacity (MB)	Purpose	Type	Speed	Slot number	Serial number
System Memory	BANK 0 (No ECC)	0	System Memory	Unknown	0	1	
System Memory	BANK 1 (No ECC)	0	System Memory	Unknown	0	2	
System Memory	BANK 2 (No ECC)	0	System Memory	Unknown	0	3	
System Memory	BANK 3 (No ECC)	4096	System Memory	DDR3	1600	4	1E6BAC89

STORAGE

2 Result(s) (Download)

Name	Manufacturer	Model	Description	Type	Disk size (MB)	Serial number	Firmware
Hitachi HDS721010CLA630 ATA Device	(Unidades de disco estándar)	//./PHYSICALDRIVE0	Unidad de disco	Fixed hard disk media	953867	JP2940N03T4JBV	JP40A41A
hp DVD A DH16ACSHR ATA Device	(Unidades de CD-ROM estándar)	hp DVD A DH16ACSHR ATA Device	Unidad de CD-ROM	DVD Writer	0		

DISK(S)

2 Result(s) (Download)

Letter	Type	Designation	File System	Free (MB)	Total (MB)	Capacity
C:	Hard Drive		NTFS	844391	953867	11%
D:	CD-Rom Drive			0	0	

Description	Type	Speed	MAC address	Status	IP address	Netmask	Gateway	Network number	DHCP IP
Intel(R) 82579LM Gigabit Network Connection	Ethernet	100 Mb/s	24-BE-05-02-BD-AA	Up	172.22.150.151	255.255.255.128	172.22.150.129	172.22.150.128	

CONTROLLER(S)

6 Result(s) (Download)

Manufacturer	Name	Type	Caption	Description	Version
(Tarjetas controladoras estándar IDE ATA/ATAPI)	ATA Channel 0	IDE Controller	ATA Channel 0	Canal IDE	
(Tarjetas controladoras estándar IDE ATA/ATAPI)	ATA Channel 2	IDE Controller	ATA Channel 2	Canal IDE	
Intel	Intel(R) 7 Series/C216 Chipset Family SATA AHCI Controller - 1E02	IDE Controller	Intel(R) 7 Series/C216 Chipset Family SATA AHCI Controller - 1E02	Intel(R) 7 Series/C216 Chipset Family SATA AHCI Controller - 1E02	
Intel	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E26	USB Controller	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E26	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E26	
Intel	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E2D	USB Controller	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E2D	Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E2D	
Intel(R) Corporation	Controladora de host Intel(R) USB 3.0 eXtensible	USB Controller	Controladora de host Intel(R) USB 3.0 eXtensible	Controladora de host Intel(R) USB 3.0 eXtensible	

Elaborado por: Livardi Salgado & Daniel Sierra

5.3 Pruebas carga y estrés

Este tipo de pruebas permiten conocer el desempeño de la aplicación en un ambiente de numerosos usuarios, las pruebas se diferencian por:

De carga: Determina y valida la respuesta de la aplicación al momento de aplicarla a una cantidad de usuarios o transacciones que se espera se supere en el ambiente de producción.

De rendimiento: Se realiza con el fin de medir la respuesta o el desempeño a distintos volúmenes de carga es decir de peticiones a la vez.

De Estrés: Determina el volumen de datos o el tiempo en que la aplicación sufre una caída o es incapaz de resolver más peticiones que se diferencia a las anteriores ya que son volúmenes extensos de peticiones a la vez.


Software necesario: Web Server Stress Tool

Edición: 7.2.2.261

5.4 Test de configuración

Durante el test de configuración se identifica las especificaciones de configuración básica, configuraciones web, datos del sistema.

Figura 41. Test de configuración de Webserver Stress tool


 Webserver Stress Tool	
Test Report	Project and Scenario Comments, Operator
Settings	Test Setup
Test Configuration	Test Type: CLICKS (run test until 1 clicks per user)
URLs	User Simulation: 5 simultaneous users - 20 seconds between clicks
Results and Graphs	Logging Period: Log every 10 seconds
Click Times and Errors	URLs
Click Time, Hits/s & Clicks/s	URL Sequencing: Users always click the same URL (to spreads load evenly on all URLs, set number of users to a multiple of the number of URLs!)
Protocol Times	URLs: Click here
Hierarchy	Browser Settings
Open Requests & Traffic	Browser Simulation: User Agent: Mozilla/5.0 (compatible; Webserver Stress Tool 7; Windows)
Test Client Health	Browser Simulation: HTTP Request Timeout: 120 s
Volume & Bandwidth	Options
Click Time Spectrum	Logging: Write detailed log(s)
Results per User	Timer: not enabled
Results per URL	Local IPs: Automatic URL#1: GET http://localhost:8082/proyectopesillo/Login.jsp POSTDATA= Click Delay=0
	Client System
	System: Windows NT V6.1 (Build 7601) Service Pack 1, CPU Proc. Lev. 686 (Rev. 15363) at 2394 MHz,
	Memory: 4294 MB available RAM of 4294 MB total physical RAM, 4294 MB available pagefile, 77481 MB free disk space on C:
	Test Software
	Webserver Stress Tool: 7.2.2.261 Trial Version

Elaborado por: Livardi Salgado & Daniel Sierra

URL identificada

La dirección web del sistema es <http://localhost:8082/proyectopesillo/Login.jsp>, el cual permite el ingreso al sistema de facturación.

Figura 42. Detección de URL




PASSLER

Webserver Stress Tool

Test Report

URL#	Name	Click Delay [s]	URL
1	Proyecto Pesillo	0	http://localhost:8082/proyectopesillo/Login.jsp



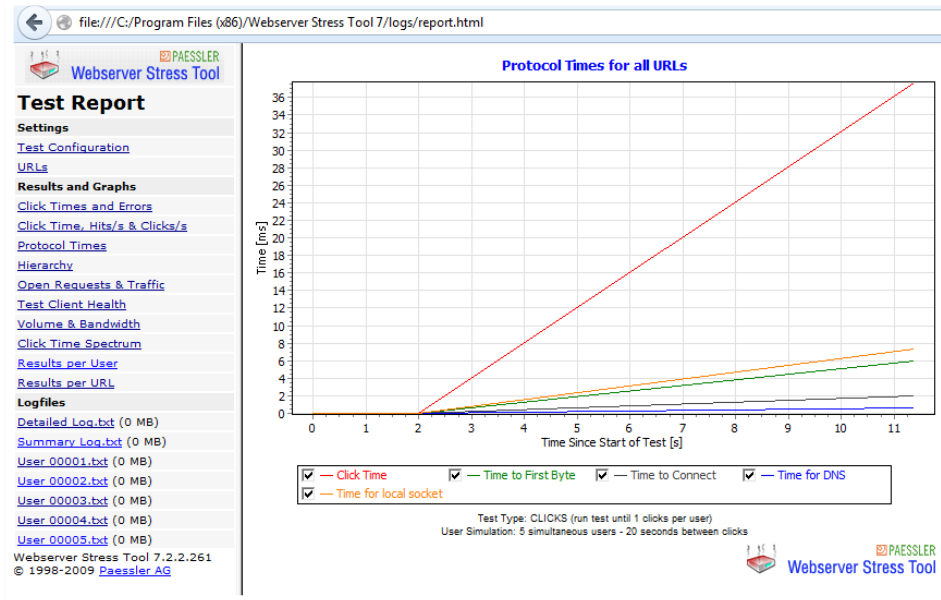
File:///C:/Program Files (x86)/Webserver Stress Tool 7/logs/report.html

Elaborado por: Livardi Salgado & Daniel Sierra

Tiempos de solución de protocolo

En la figura 43, se observa el rendimiento al realizar conexión hacia el servidor por parte de los equipos clientes.

Figura 43. Tiempos de detección de protocolos

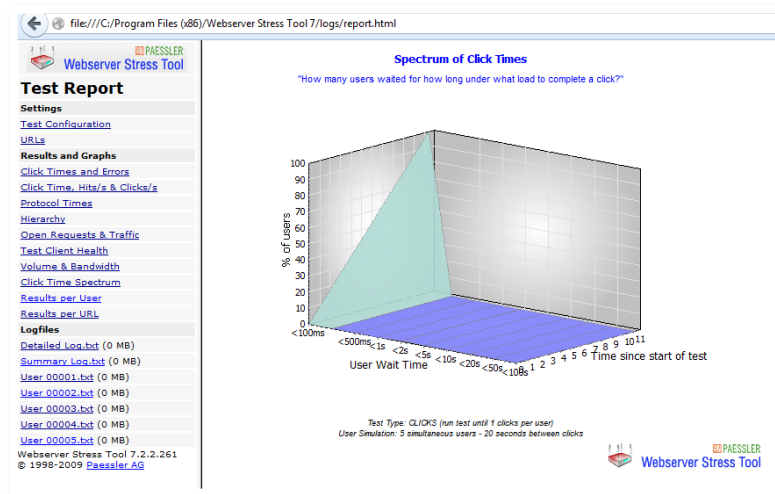


Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Espectro

En la figura 44 se identifica el espectro en relación a la carga y rendimiento del sistema al potenciarlo en su carga.

Figura 44. Análisis de carga en espectro

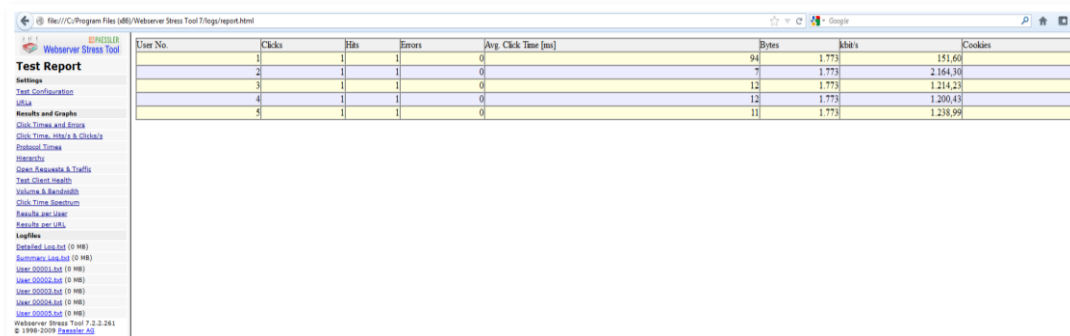


Elaborado por: Livardi Salgado, Daniel Sierra

Resultados por usuario

En la figura 45, se identifica la cantidad de usuarios en base a la carga obtenida al realizar la conexión correspondiente al servidor.

Figura 45. Resultados de Gestión Usuarios



The screenshot shows a web browser window displaying a report from the Webserver Stress Tool. The report is titled 'Test Report' and contains a table with the following data:

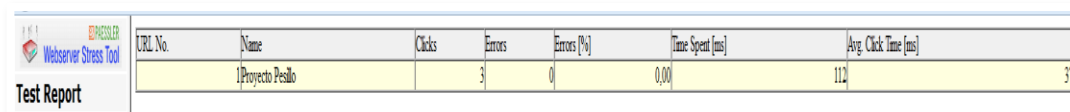
User No.	Clicks	Hits	Errors	Avg. Click Time (ms)	Bytes	kbits	Cookies
1	1	1	0	0	94	1.773	151.66
2	1	1	0	0	7	1.773	2.164.38
3	1	1	0	0	12	1.773	1.214.23
4	1	1	0	0	12	1.773	1.200.43
5	1	1	0	0	11	1.773	1.238.99

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados por URL

En la figura 46, determina la carga en base a la cantidad de usuarios soportados para la dirección URL.

Figura 46. Resultado de conexión a URL



The screenshot shows a web browser window displaying a report from the Webserver Stress Tool. The report is titled 'Test Report' and contains a table with the following data:

URL No.	Name	Clicks	Errors	Errors (%)	Time Spent (ms)	Avg. Click Time (ms)
1	Proyecto Pésalo	3	0	0.00	112	37

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Equipo 1

En la figura 47, muestra la siguiente información:

- Tiempo en conectarse

- Identificación de servicio
- Indicador de conexión
- Se realiza peticiones básicas
- Cierra conexión
- Entrega resultados

Figura 47. Detalle de resultado de Equipo 1

```

Webserver Stress Tool - Logfile for User #1
18/05/2014 19:37:43: User #1 Click #1: Waiting for global click delay: 20s
18/05/2014 19:37:43: User #1 Click #1: HTTP request & TCP/IP socket creation: 8 ms
18/05/2014 19:37:43: User #1 Click #1: ** Click #1 started, used URL #1 (Proyecto Pesillo), from IP=0.0.0.0
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Initialized (0 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: DNS Lookup begin (0 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: DNS Lookup done (1 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Connected (3 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Waiting for header (3 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Receiving Header (10 ms) (=Time to First Byte)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Waiting for body (10 ms)
18/05/2014 19:37:43: User #1 Click #1: Request 1 HTTP State: Done (88 ms)
18/05/2014 19:37:43: User #1 Click #1: CLICK-Request 1: Time=93 ms, TFB=10 ms, Bytes=1.773, HTTP-StatusCode=200 (OK)
18/05/2014 19:37:43: User #1 Click #1: !Success! All requests of click done (94 ms)
18/05/2014 19:38:00: Writing User Log
18/05/2014 19:38:01: User passed away

```

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Equipo 2

En la figura 48, muestra la siguiente información:

- Tiempo en conectarse
- Identificación de servicio
- Indicador de conexión
- Se realiza peticiones básicas
- Cierra conexión
- Entrega resultados

Figura 48. Detalle de resultado de Equipo 2

```
Webserver Stress Tool - Logfile for User #2
18/05/2014 19:37:43: User #2 Click #1: Waiting for global click delay: 20s
18/05/2014 19:37:47: User #2 Click #1: HTTP request & TCP/IP socket creation: 9 ms
18/05/2014 19:37:47: User #2 Click #1: ** Click #1 started, used URL #1 (Proyecto Pesillo), from IP=0.0.0.0
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Initialized (0 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: DNS Lookup begin (0 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: DNS Lookup done (0 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Connected (1 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Waiting for header (1 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Receiving Header (3 ms) (=Time to First Byte)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Waiting for body (3 ms)
18/05/2014 19:37:47: User #2 Click #1: Request 1 HTTP State: Done (4 ms)
18/05/2014 19:37:47: User #2 Click #1: CLICK-Request 1: Time=6 ms, TFB=3 ms, Bytes=1.773, HTTP-StatusCode=200 (OK)
18/05/2014 19:37:47: User #2 Click #1: !Success! All requests of click done (7 ms)
18/05/2014 19:38:00: Writing User Log
18/05/2014 19:38:01: User passed away
```

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Equipo 3

En la figura 49, muestra la siguiente información:

- Tiempo en conectarse
- Identificación de servicio
- Indicador de conexión
- Se realiza peticiones básicas
- Cierra conexión
- Entrega resultados

Figura 49. Detalle de resultado de Equipo 3

```
Webserver Stress Tool - Logfile for User #3
18/05/2014 19:37:43: User #3 Click #1: Waiting for global click delay: 20s
18/05/2014 19:37:51: User #3 Click #1: HTTP request & TCP/IP socket creation: 5 ms
18/05/2014 19:37:51: User #3 Click #1: ** Click #1 started, used URL #1 (Proyecto Pesillo), from IP=0.0.0.0
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Initialized (0 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: DNS Lookup begin (0 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: DNS Lookup done (1 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Connected (2 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Waiting for header (2 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Receiving Header (5 ms) (=Time to First Byte)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Waiting for body (5 ms)
18/05/2014 19:37:51: User #3 Click #1: Request 1 HTTP State: Done (6 ms)
18/05/2014 19:37:51: User #3 Click #1: CLICK-Request 1: Time=12 ms, TFB=5 ms, Bytes=1.773, HTTP-StatusCode=200 (OK)
18/05/2014 19:37:51: User #3 Click #1: !Success! All requests of click done (12 ms)
18/05/2014 19:38:00: Writing User Log
18/05/2014 19:38:01: User passed away
```

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Equipo 4

En la figura 50, muestra la siguiente información:

- Tiempo en conectarse
- Identificación de servicio
- Indicador de conexión
- Se realiza peticiones básicas
- Cierra conexión
- Entrega resultados

Figura 50. Detalle de resultado de Equipo 4:

```
Webserver Stress Tool - Logfile for User #4
18/05/2014 19:37:43: User #4 Click #1: Waiting for global click delay: 20s
18/05/2014 19:37:55: User #4 Click #1: HTTP request & TCP/IP socket creation: 6 ms
18/05/2014 19:37:55: User #4 Click #1: ** Click #1 started, used URL #1 (Proyecto Pesillo), from IP=0.0.0.0
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Initialized (0 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: DNS Lookup begin (0 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: DNS Lookup done (1 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Connected (2 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Waiting for header (2 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Receiving Header (6 ms) (=Time to First Byte)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Waiting for body (6 ms)
18/05/2014 19:37:55: User #4 Click #1: Request 1 HTTP State: Done (6 ms)
18/05/2014 19:37:55: User #4 Click #1: CLICK-Request 1: Time=12 ms, TFB=6 ms, Bytes=1.773, HTTP-StatusCode=200 (OK)
18/05/2014 19:37:55: User #4 Click #1: !Success! All requests of click done (12 ms)
18/05/2014 19:38:00: Writing User Log
18/05/2014 19:38:01: User passed away
```

Elaborado por: Livardi Salgado & Daniel Sierra

Resultados de Equipo 5

En la figura 51, muestra la siguiente información:

- Tiempo en conectarse
- Identificación de servicio
- Indicador de conexión
- Se realiza peticiones básicas
- Cierra conexión
- Entrega resultados

Figura 51. Detalle de resultado de Equipo 5

```
Webserver Stress Tool - Logfile for User #5
18/05/2014 19:37:43: User #5 Click #1: Waiting for global click delay: 20s
18/05/2014 19:37:59: User #5 Click #1: HTTP request & TCP/IP socket creation: 6 ms
18/05/2014 19:37:59: User #5 Click #1: ** Click #1 started, used URL #1 (Proyecto Pesillo), from IP=0.0.0.0
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Initialized (0 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: DNS Lookup begin (0 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: DNS Lookup done (1 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Connected (2 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Waiting for header (2 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Receiving Header (5 ms) (=Time to First Byte)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Waiting for body (5 ms)
18/05/2014 19:37:59: User #5 Click #1: Request 1 HTTP State: Done (6 ms)
18/05/2014 19:37:59: User #5 Click #1: CLICK-Request 1: Time=11 ms, TFB=5 ms, Bytes=1.773, HTTP-StatusCode=200 (OK)
18/05/2014 19:37:59: User #5 Click #1: !Success! All requests of click done (11 ms)
18/05/2014 19:38:00: Writing User Log
18/05/2014 19:38:01: User passed away
```

Elaborado por: Livardi Salgado & Daniel Sierra

CONCLUSIONES

Por medio de pruebas de estrés e implementación del sistema junto a los aplicativos necesarios para su funcionamiento, se concluye que los recursos utilizados para el procesamiento de información son mínimos por tanto su funcionamiento es óptimo en un ordenador que cumple las características de hardware recomendadas.

La facilidad del desarrollo en JSP y el beneficio de portabilidad entre plataformas y el lenguaje orientado a objetos que ofrece para la generación de contenido dinámico.

La utilización de una base de datos robusta y que permita aumentar su tamaño, que mantenga su estabilidad y confiabilidad es de gran beneficio para cualquier aplicación. Por esa razón se eligió Postgresql al cumplir estos parámetros.

El beneficio de utilizar código abierto y herramientas libres aportó a la elaboración del sistema, adaptación a las necesidades e información disponible para el desarrollo.

Durante la implementación se analizó las diferentes herramientas de reportes y se eligió a IREPORT debido a su desarrollo basado en Java, a sus herramientas de múltiples diseños y los asistentes de plantillas que permite integrar fácilmente a nuestro sistema.

Con respecto a la implementación, se concluye la entrega de un sistema sólido, estable y modificable para futuras necesidades, el cual puede ser utilizado para diferentes proyectos con beneficio común.

Es necesario entregar la información necesaria para la instalación, modificación y aprendizaje del sistema entregado, adjuntando los recursos para su funcionamiento, manuales técnicos para futuros avances y manual de usuario digital y videos útiles del proceso de instalación y uso del mismo, logrando cumplir los objetivos planteados para el desarrollo de la presente tesis de graduación.

RECOMENDACIONES

Es de gran utilidad durante el desarrollo del sistema, realizar las copias de seguridad necesarias del diseño e implementación con el fin respaldar la información tanto del código fuente como de la base datos, este paso permite realizar una regresión en el caso de ser necesario, este proceso se puede optimizar con una tarea programada mediante Windows.

Es de gran utilidad realizar pruebas en diferentes versiones de navegadores con el fin de entregar un sistema multiversión debido a las actualizaciones de la plataforma.

El uso de aplicaciones web permite obtener la información de una manera dinámica y de fácil interpretación para el cliente, cabe mencionar que se debe obtener los requerimientos del sistema básicos para su ejecución y funcionalidad.

Es útil que durante el avance con el desarrollo del sistema realizar las pruebas y que las mismas se ejecuten por el usuario final, con la finalidad de observar los errores que puedan tener en el sistema, la complejidad o errores involuntarios del sistema al procesar el sistema, con ello se evidencia las preguntas frecuentes útiles a los usuarios.

Se debe llevar a cabo pruebas de estrés en el equipo donde se instalará los aplicativos y herramientas que el sistema usará, con ello se va registrando el uso y desempeño del sistema en el equipo instalado.

Se recomienda antes de la utilización del sistema la lectura y visualización de los manuales de uso escrito y del video con el fin de obtener una primera percepción del sistema que permitirá al usuario una adaptación más sutil al sistema.

LISTA DE REFERENCIAS

- dokuwiki. (2013). *www.programacionfacil.com*. Obtenido de http://www.programacionfacil.com/java_jsp/start
- Javalea Blog de Informática. (2009). Obtenido de <http://javalea.wordpress.com/2009/01/09/conexion-jsp-postgresql/>
- Mestras, J. P. (2013). *Java EE – JavaServer Pages*. Obtenido de Aplicaciones Web/Sistemas Web: fesor/jpavon/web/44-jsp.pdf
- TI, S. (22 de 07 de 2012). *Java Mexico*. Obtenido de http://www.javamexico.org/blogs/roger1345/consultado_registros_de_una_bd_usando_jsp_y_servlets
- Brett D. McLaughlin, G. P. (2006). *Head First Object Oriented Analysis and Design*. ISBN.
- dokuwiki. (2013). *www.programacionfacil.com*. Obtenido de http://www.programacionfacil.com/java_jsp/start
- Eckel, B. (2006). *Thinking in Java*. ISBN.
- Etheridge, D. (2003). *Java: Graphical User Interfaces*. ISBN.
- James Gosling, B. J. (2013). *The Java Language Specification*. ISBN-10.
- Javalea Blog de Informática. (2009). Obtenido de <http://javalea.wordpress.com/2009/01/09/conexion-jsp-postgresql/>
- Kathy Sierra, B. B. (2005). *Head First Java*. ISBN-10 .
- Kendal, S. (2005). *Object Oriented Programming using Java*. ISBN.
- Kiet T. Tran, P. (2007). *Introduction to Web Services with Java*. ISBN.
- Maria, S. M. (2003). *Metodologia de desarrollo de software*.
- Mestras, J. P. (2013). *Java EE – JavaServer Pages*. Obtenido de Aplicaciones Web/Sistemas Web: fesor/jpavon/web/44-jsp.pdf
- Sánchez, M. (2003). *Metodología de desarrollo de software*.

Sharon Biocca Zakhour, S. K. (2013). *A Short Course on the Basics*. ISBN-10.

TI, S. (22 de 07 de 2012). *Java Mexico*. Obtenido de
http://www.javamexico.org/blogs/roger1345/consultado_registros_de_una_bd_usando_jsp_y_servlets

Anexo 1. Diagramas múltiples

ÍNDICE

DIAGRAMAS MULTIPLES	1
1.1 Módulo autenticación	1
1.1.1 Proceso autenticación.	1
1.2 Modelo usuario.....	2
1.2.1 Proceso creación de usuario.	2
1.2.2 Proceso creación de usuario.	4
1.2.3 Proceso eliminación de usuario.....	5
1.3 Modelo administrador	6
1.3.1 Proceso creación.....	6
1.3.2 Proceso actualización de administrador.	7
1.3.3 Proceso eliminación de clientes.....	8
1.4 Modelo de clientes	9
1.4.1 Proceso creación.....	9
1.4.2 Proceso actualización de clientes.	10
1.4.3 Proceso eliminación de clientes.....	11
1.5 Modelo perfiles	12
1.6 Modelo medidor	13
1.6.1 Proceso creación.....	13
1.6.2 Proceso actualización medidor.	14
1.6.3 Proceso eliminación de clientes.....	15
1.7 Modelo lectura de información.....	16
1.7.1 Proceso registro.....	16
1.8 Modelo eventos	17
1.8.1 Proceso creación.....	17
1.9 Modelo multas	18
1.10 Modelo facturas.....	19
1.10.1 Proceso creación.....	19
1.11 Modelo reportes	20
1.11.1 Proceso generación.....	20

CONTENIDO

Se presenta la diagramación de todas las interfaces y funcionalidades del sistemas tanto en el modo de uso, la diagramación de secuencia y actividad de cada a evento a desarrollar

DIAGRAMAS MULTIPLES

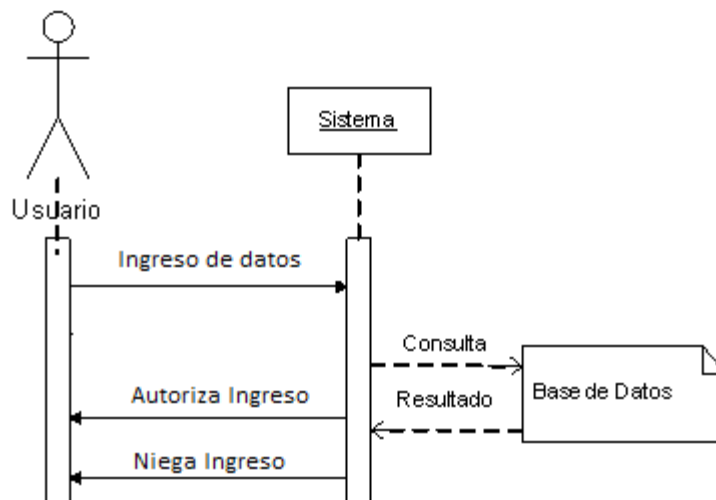
1.1 Módulo autenticación

En el presente modulo se puede identificar el acceso al sistema y sus autorizadores

1.1.1 Proceso autenticación.

Modelo de Caso de Secuencia #1 – Autenticación

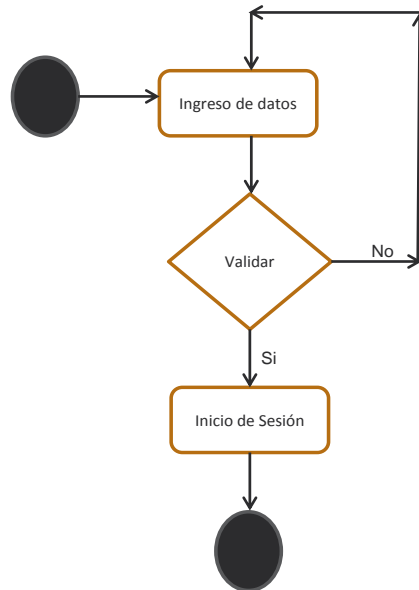
Figura 1. Modelo de Caso de Secuencia#1, Proceso Autenticación



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #1 – Autenticación

Figura 2. Modelo de Caso de Actividad, Proceso Autenticación



Elaborado por: Livardi Salgado & Daniel Sierra

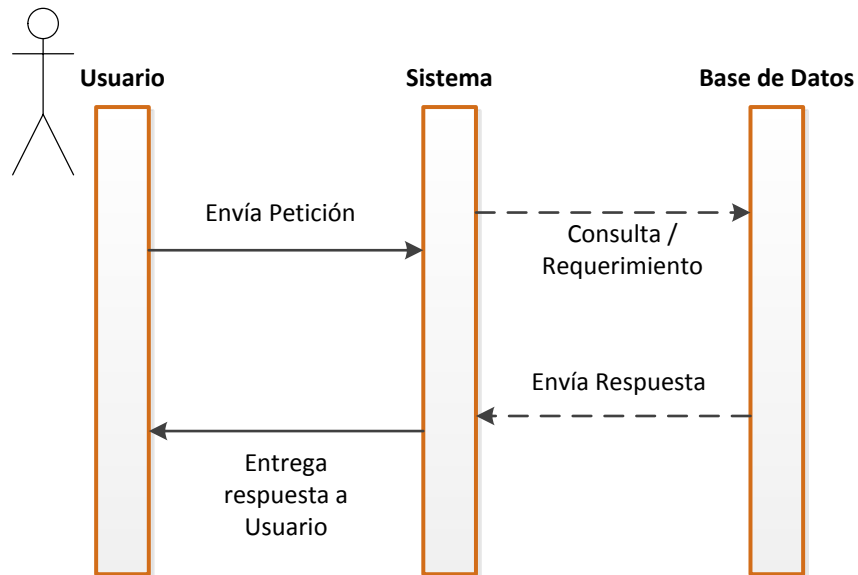
1.2 Modelo usuario

1.2.1 Proceso creación de usuario.

Se define a la creación de los diferentes usuarios que ejecuten y procesen la información del sistema (colaboradores).

Modelo de Caso de Secuencia #2– Creación de Usuario

Figura 3. Modelo de Caso de Secuencia, Proceso Creación de Usuario.



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #2– Creación de Usuario

Figura 4. Modelo de Caso de Actividad, Proceso Creación de Usuario.

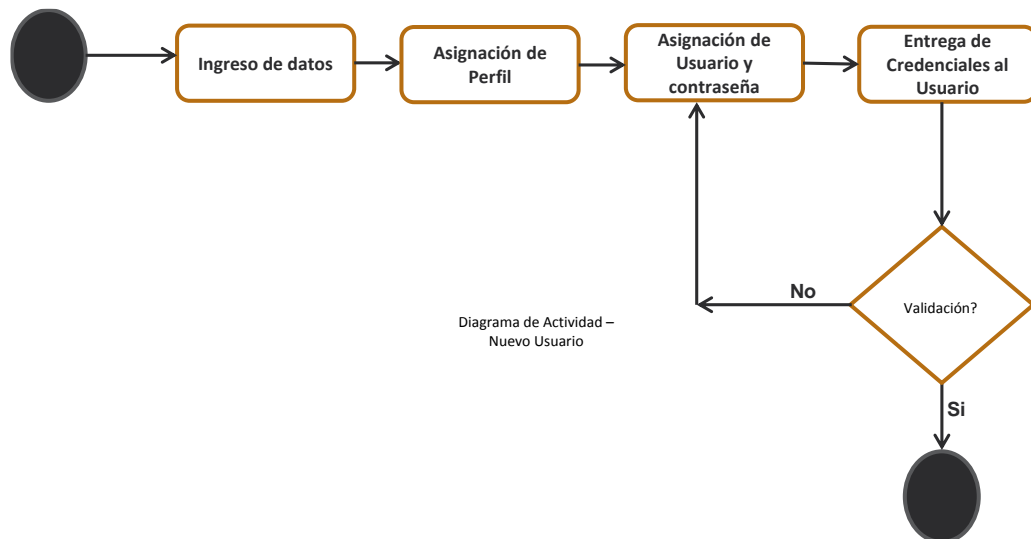
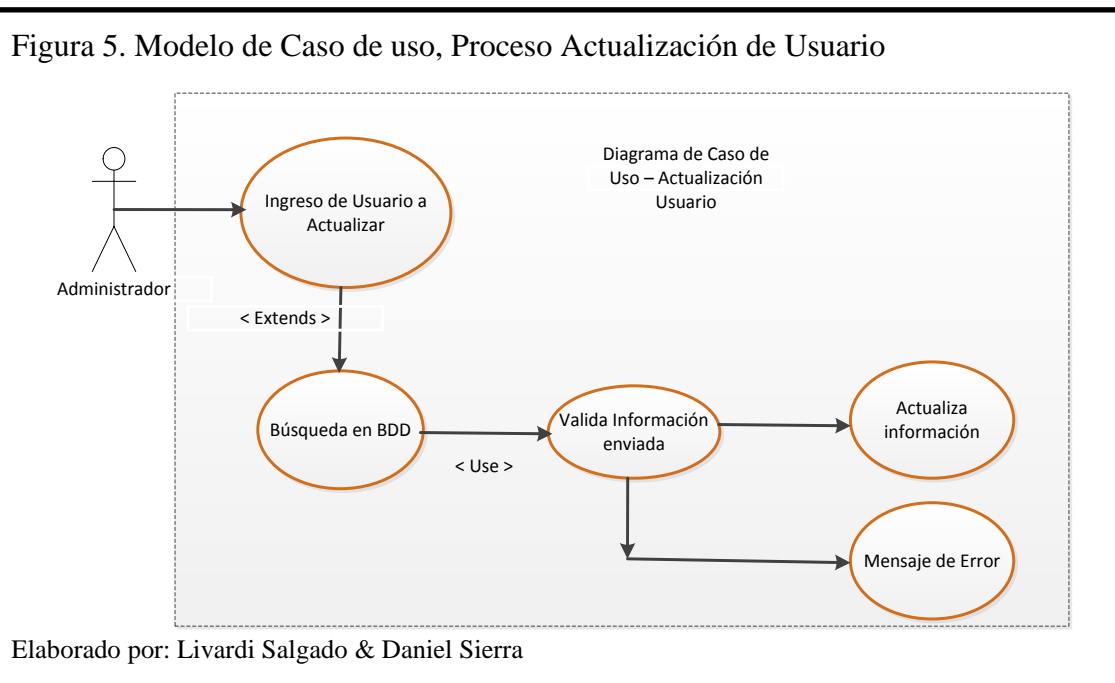


Diagrama de Actividad –
Nuevo Usuario

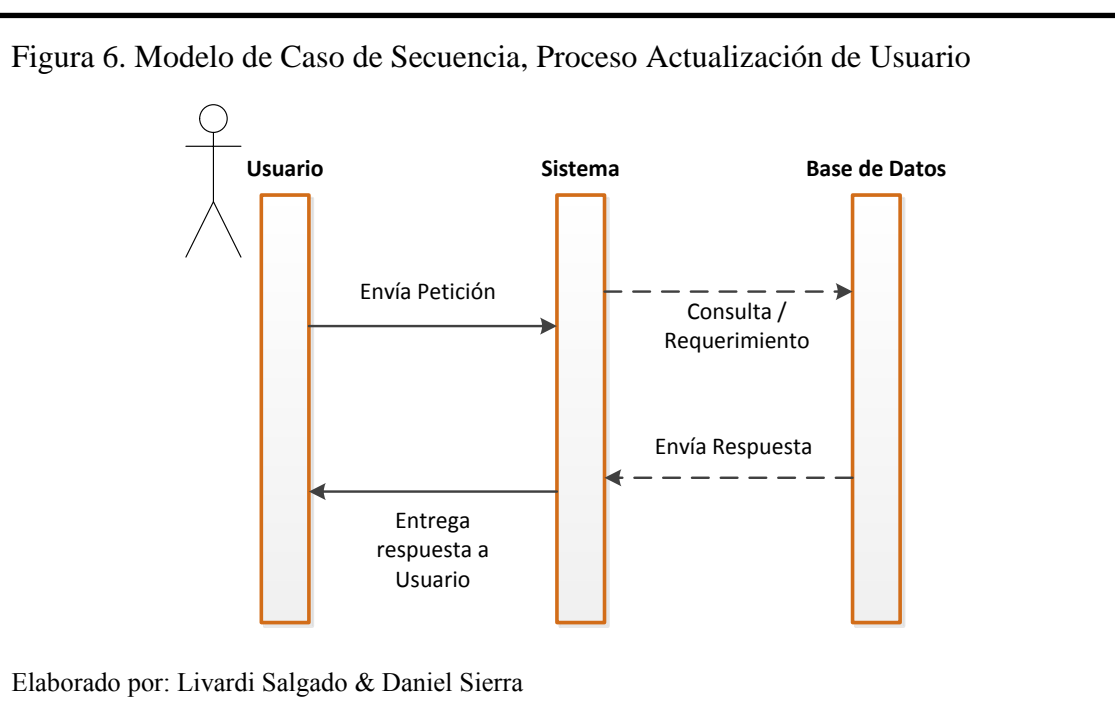
Elaborado por: Livardi Salgado & Daniel Sierra

1.2.2 Proceso creación de usuario.

Modelo de Caso de Uso #3 – Actualización de Usuario

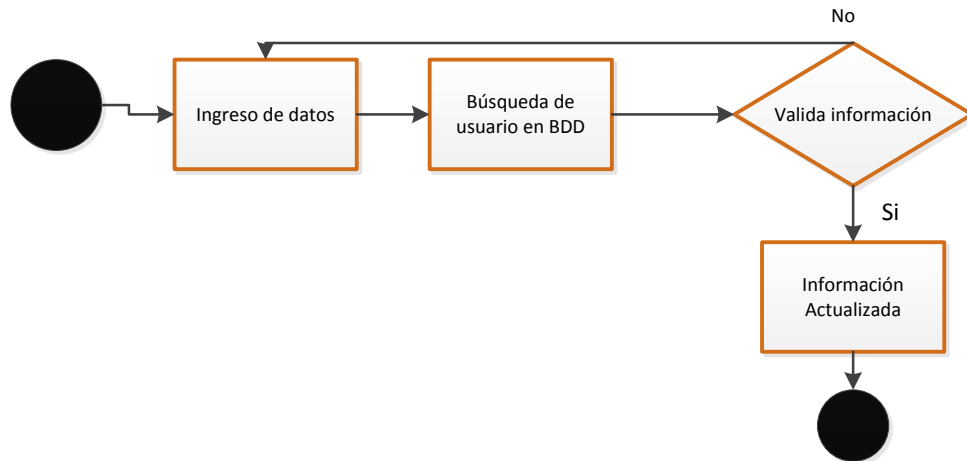


Modelo de Caso de Secuencia #3 – Actualización de Usuario



Modelo de Caso de Actividad #3 – Actualización de Usuario

Figura 7. Modelo de Caso de Actividad, Proceso Actualización de Usuario

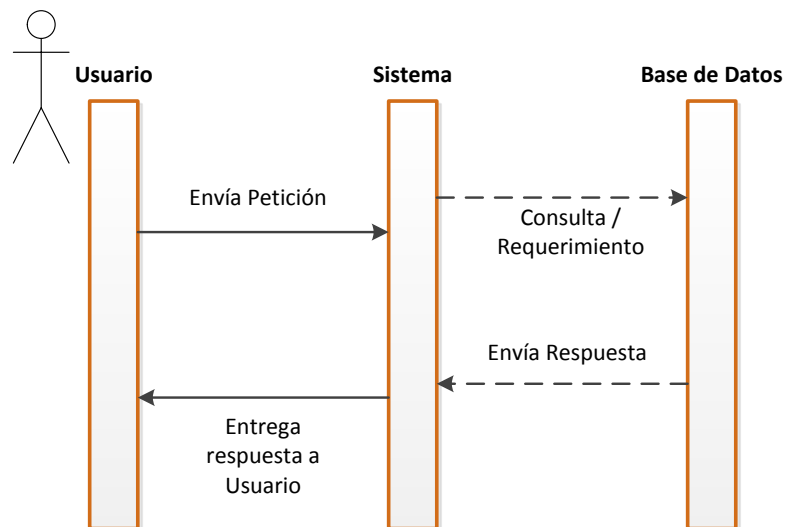


Elaborado por: Livardi Salgado & Daniel Sierra

1.2.3 Proceso eliminación de usuario.

Modelo de Caso de Secuencia #4– Eliminación de Usuario

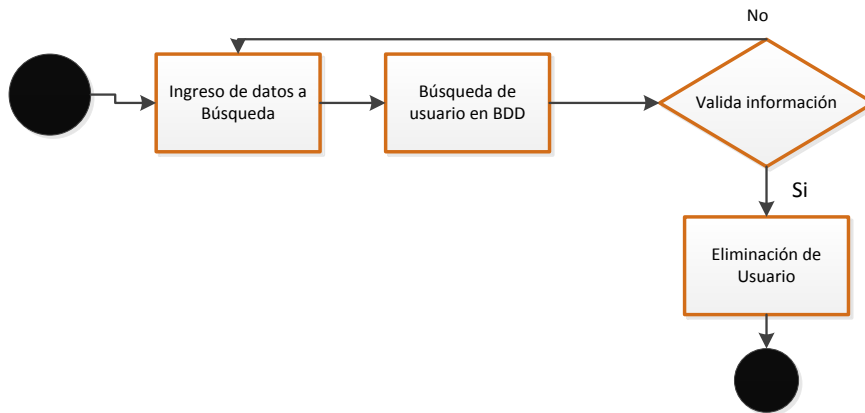
Figura 8. Modelo de Caso de Secuencia, Proceso Eliminación de Usuario



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #4– Eliminación de Usuario

Figura 9. Modelo de Caso de Actividad, Proceso Eliminación de Usuario



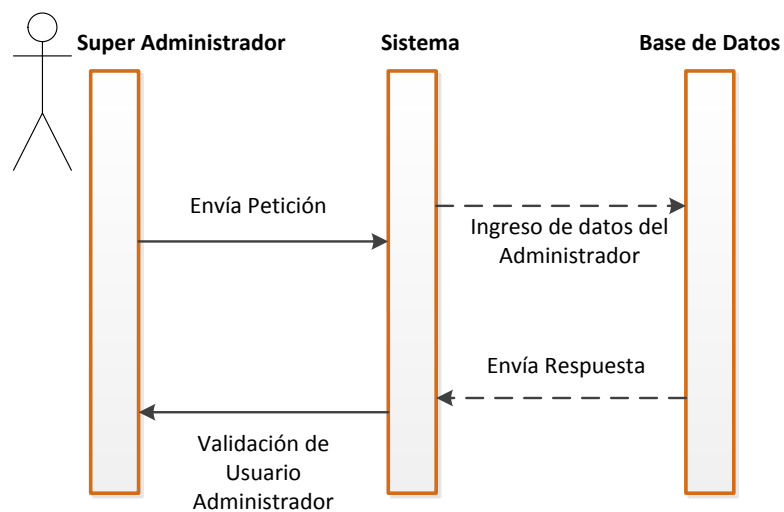
Elaborado por: Livardi Salgado & Daniel Sierra

1.3 Modelo administrador

1.3.1 Proceso creación.

Modelo de Caso de Secuencia #5– Creación Administrador

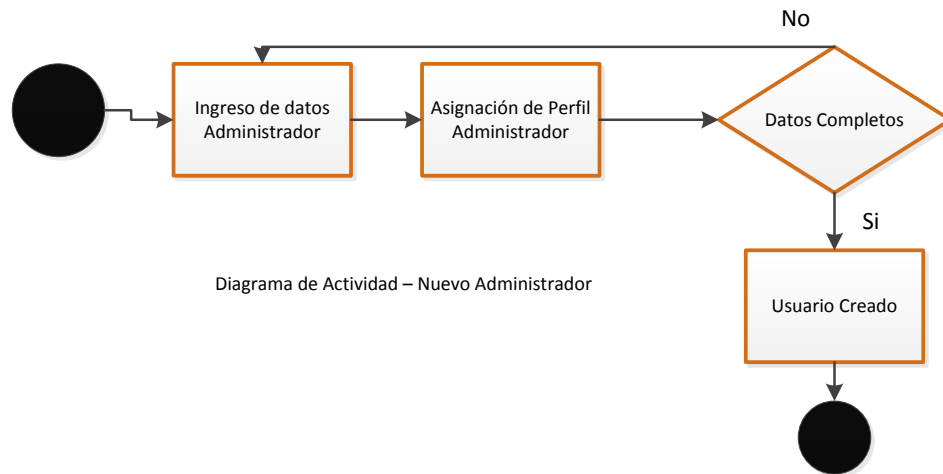
Figura 10. Modelo de Caso de Secuencia, Proceso Creación de Administrador



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #5– Creación Administrador

Figura 11. Modelo de Caso de Actividad, Proceso Creación de Administrador

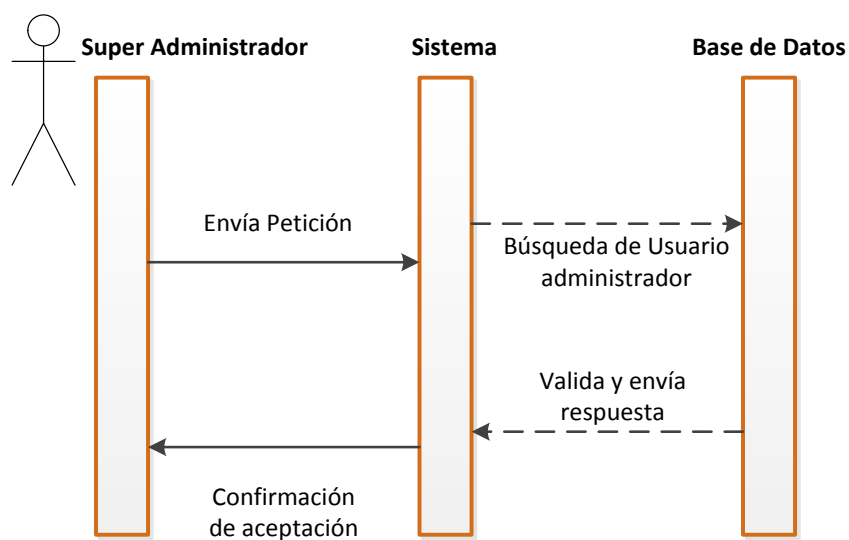


Elaborado por: Livardi Salgado & Daniel Sierra

1.3.2 Proceso actualización de administrador.

Modelo de Caso de Secuencia #6 – Actualización Administrador

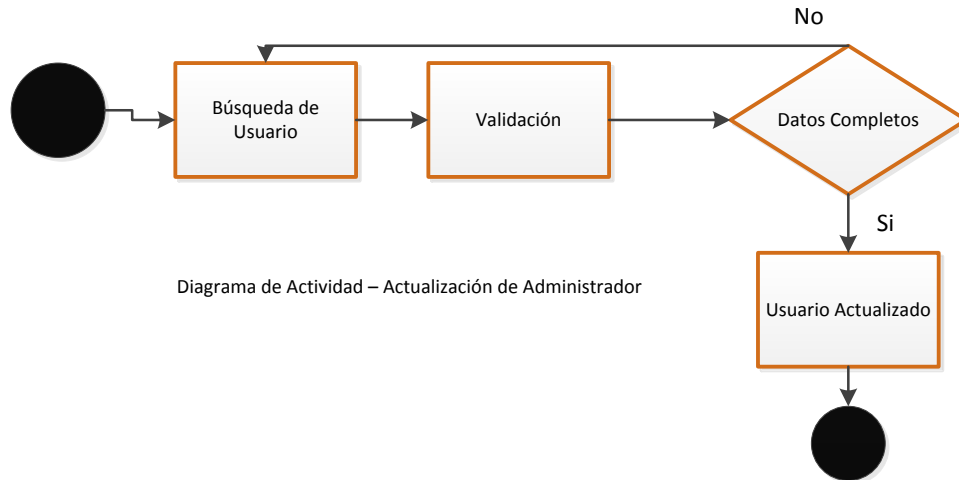
Figura 12. Modelo de Caso de Secuencia, Proceso Actualización de Administrador



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #6 – Actualización Administrador

Figura 13. Modelo de Caso de Actividad, Proceso Actualización de Administrador

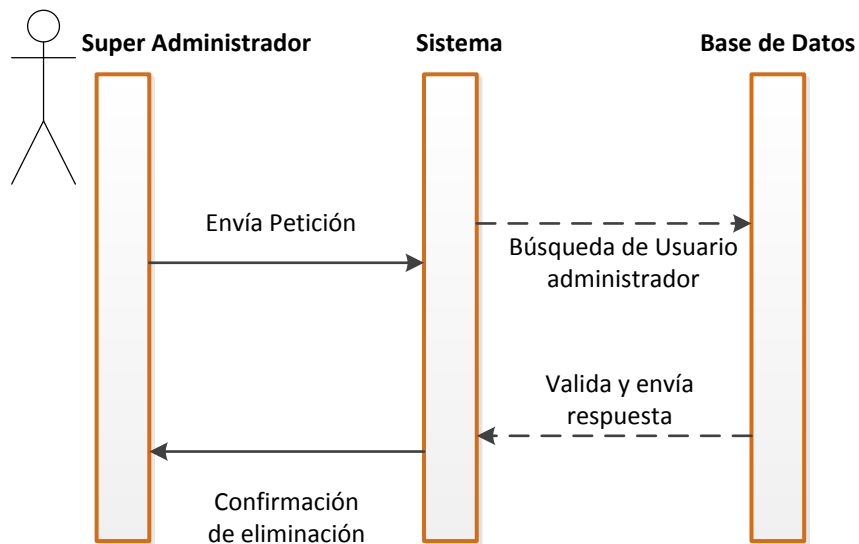


Elaborado por: Livardi Salgado & Daniel Sierra

1.3.3 Proceso eliminación de clientes.

Modelo de Caso de Secuencia #7 – Eliminación Administrador

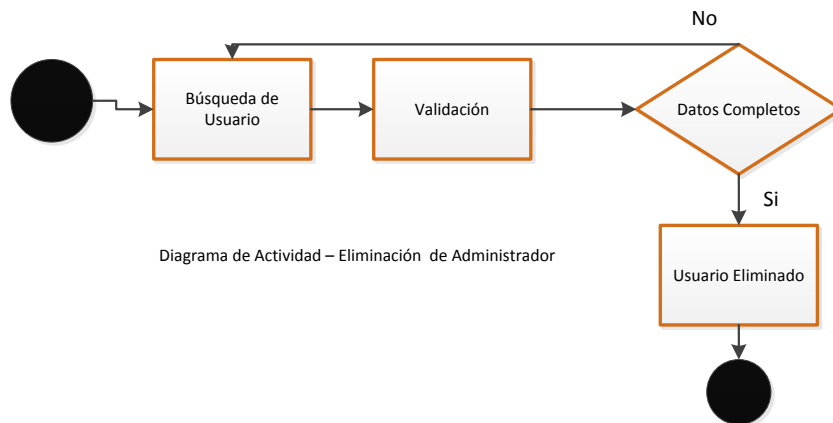
Figura 14. Modelo de Caso de Secuencia, Proceso Eliminación de Administrador



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #7 – Eliminación Administrador

Figura 15. Modelo de Caso de Actividad, Proceso Eliminación de Administrador



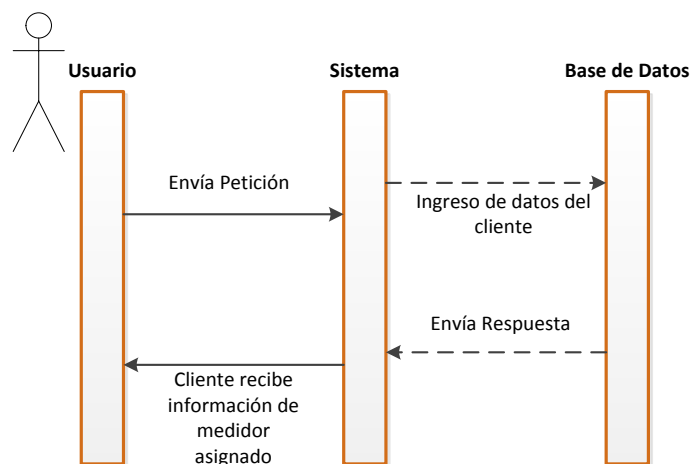
Elaborado por: Livardi Salgado & Daniel Sierra

1.4 Modelo de clientes

1.4.1 Proceso creación.

Modelo de Caso de Secuencia #8 – Creación Clientes

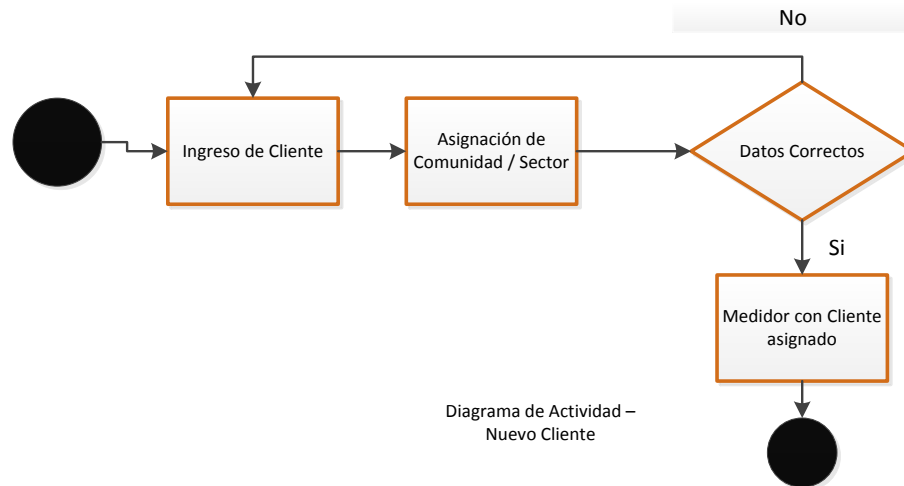
Figura 16. Modelo de Caso de Secuencia, Proceso Creación de Clientes.



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #8 – Creación Clientes

Figura 17. Modelo de Caso de Actividad, Proceso Creación de Clientes.

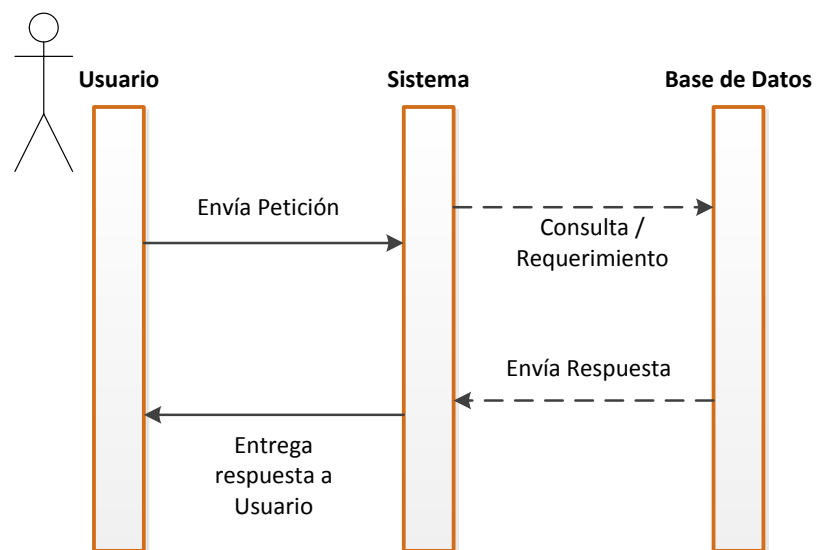


Elaborado por: Livardi Salgado & Daniel Sierra

1.4.2 Proceso actualización de clientes.

Modelo de Caso de Secuencia #9 – Actualización Cliente

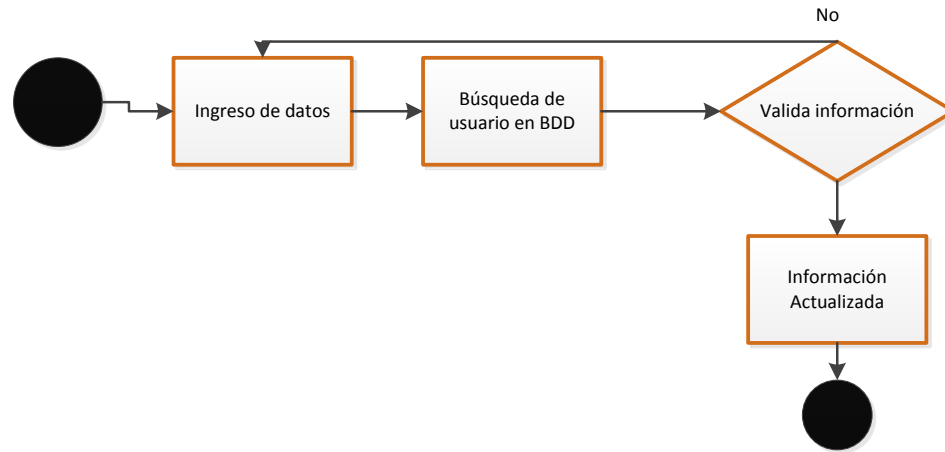
Figura 18. Modelo de Caso de Secuencia, Proceso Actualización de Clientes.



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #9 – Actualización Cliente

Figura 19. Modelo de Caso de Actividad, Proceso Actualización de Clientes

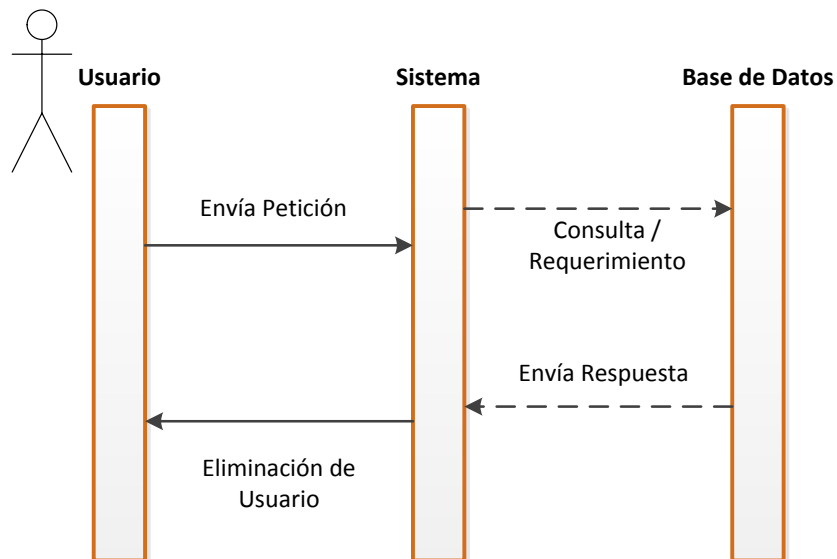


Elaborado por: Livardi Salgado & Daniel Sierra

1.4.3 Proceso eliminación de clientes.

Modelo de Caso de Secuencia #10 – Eliminación Cliente

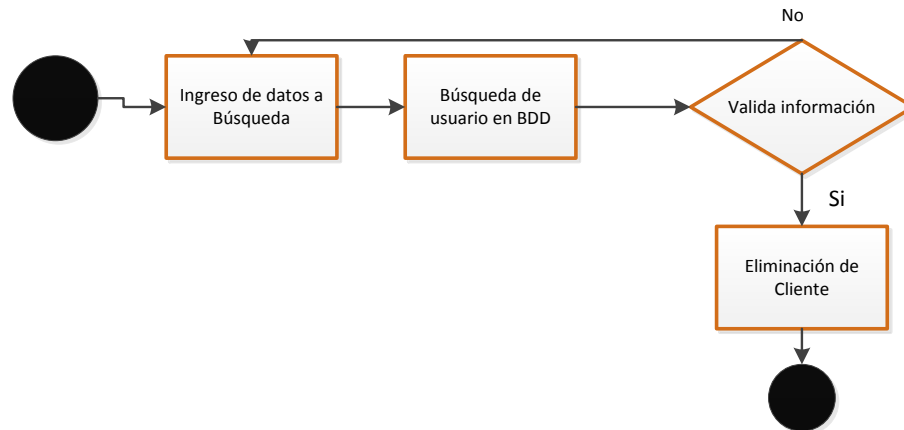
Figura 20. Modelo de Caso de Secuencia, Proceso Eliminación de Clientes



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #10 – Eliminación Cliente

Figura 21. Modelo de Caso de Actividad, Proceso Eliminación de Clientes.

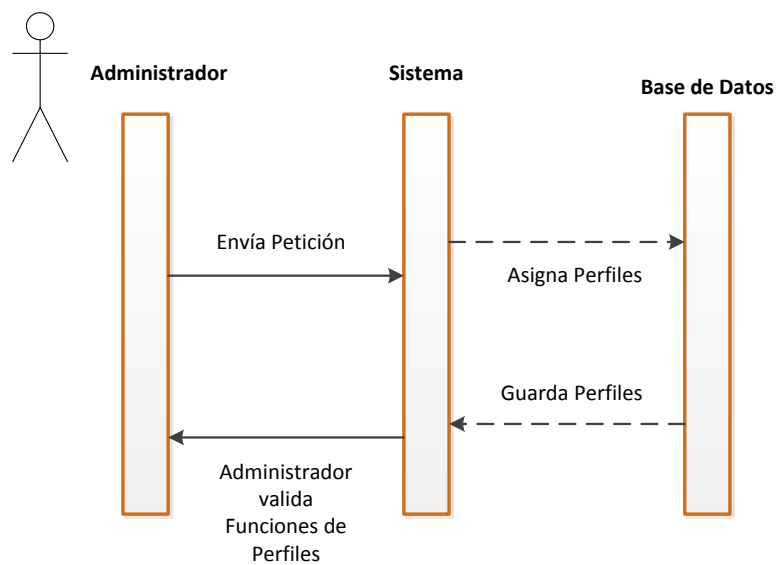


Elaborado por: Livardi Salgado & Daniel Sierra

1.5 Modelo perfiles

Modelo de Caso de Secuencia #11 – Creación Perfil

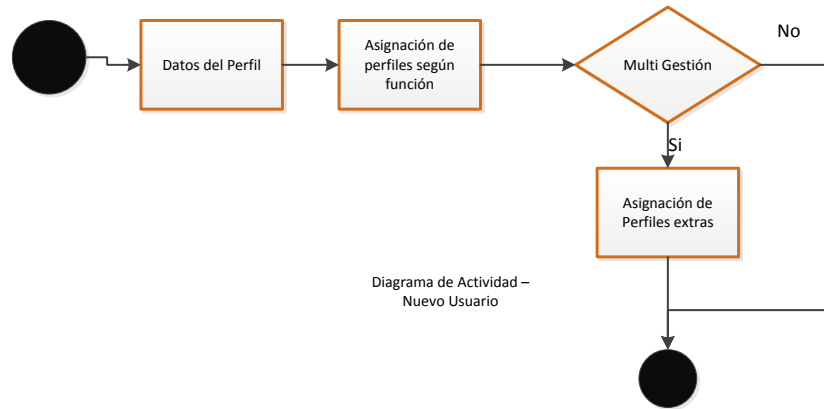
Figura 22. Modelo de Caso de Secuencia, Proceso Creación de Perfil



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #11 – Creación Perfil

Figura 23. Modelo de Caso de Actividad, Proceso Creación de Perfil



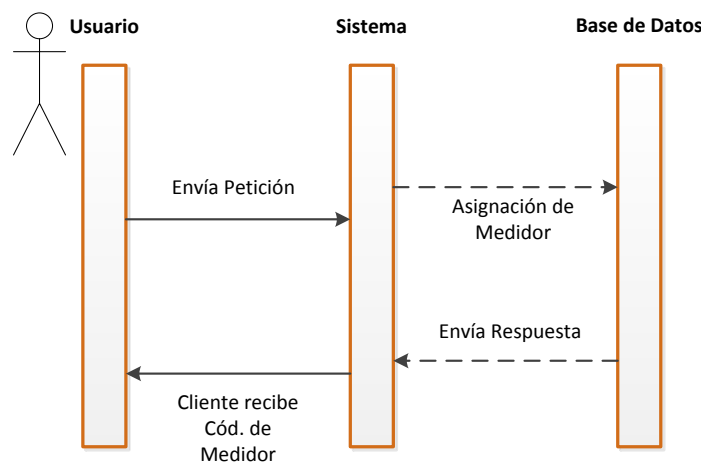
Elaborado por: Livardi Salgado & Daniel Sierra

1.6 Modelo medidor

1.6.1 Proceso creación.

Modelo de Caso de Secuencia #12 – Creación Medidor

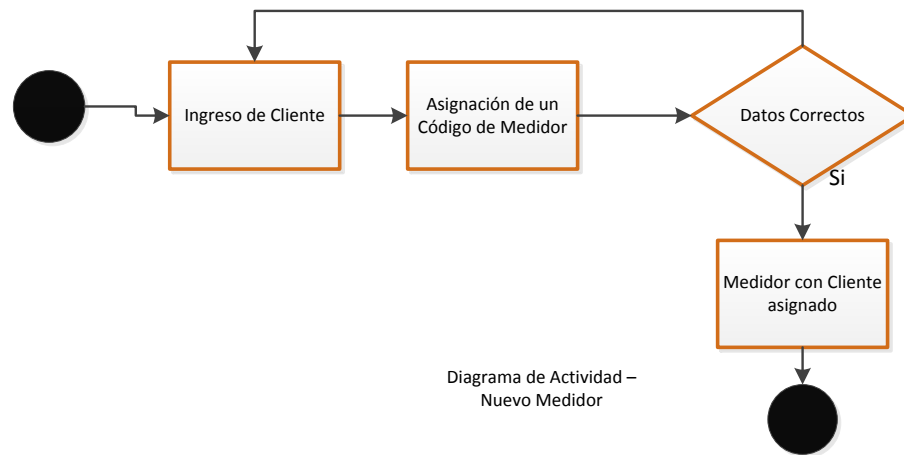
Figura 24. Modelo de Caso de Secuencia, Proceso Creación de Medidor



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #12 – Creación Medidor

Figura 25. Modelo de Caso de Actividad, Proceso Creación de Medidor

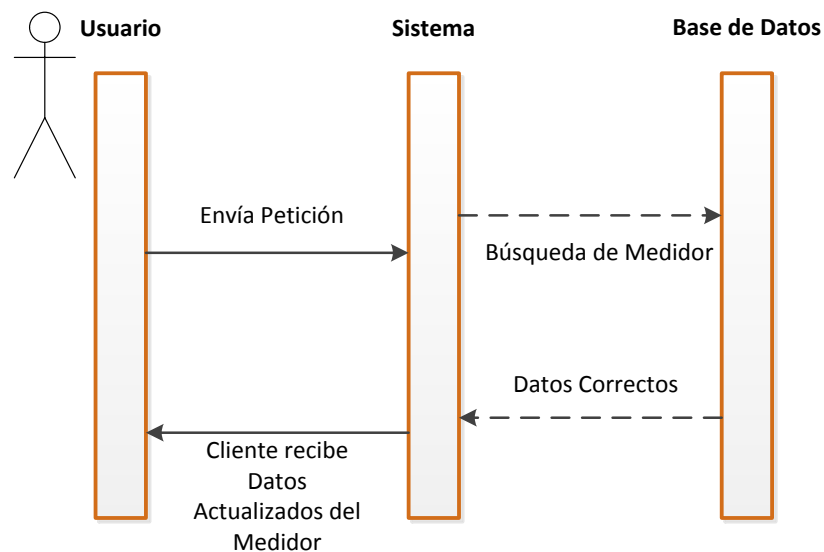


Elaborado por: Livardi Salgado & Daniel Sierra

1.6.2 Proceso actualización medidor.

Modelo de Caso de Secuencia #13 – Actualización Medidor

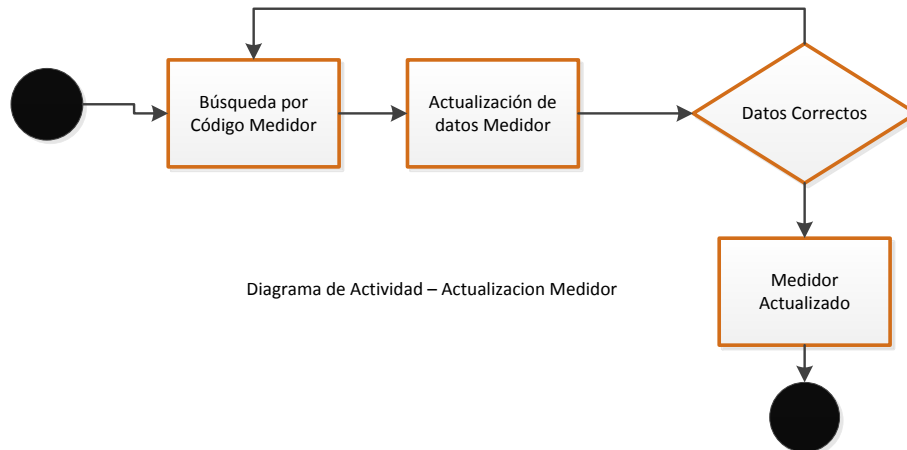
Figura 26. Modelo de Caso de Secuencia, Proceso Actualización de Medidor



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #13 – Actualización Medidor

Figura 27. Modelo de Caso de Actividad, Proceso Actualización de Medidor

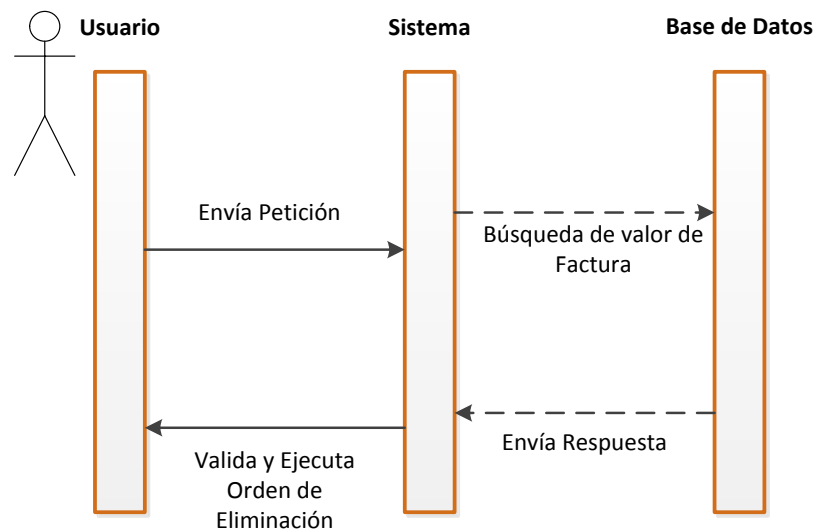


Elaborado por: Livardi Salgado & Daniel Sierra

1.6.3 Proceso eliminación de clientes.

Modelo de Caso de Secuencia #14 – Eliminación Medidor

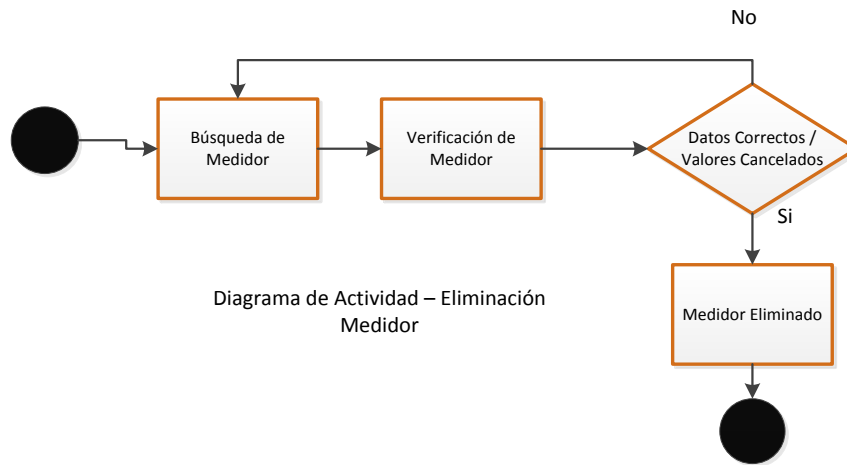
Figura 28. Modelo de Caso de Secuencia, Proceso Eliminación de Medidor



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #14 – Eliminación Medidor

Figura 29. Modelo de Caso de Actividad, Proceso Eliminación de Medidor



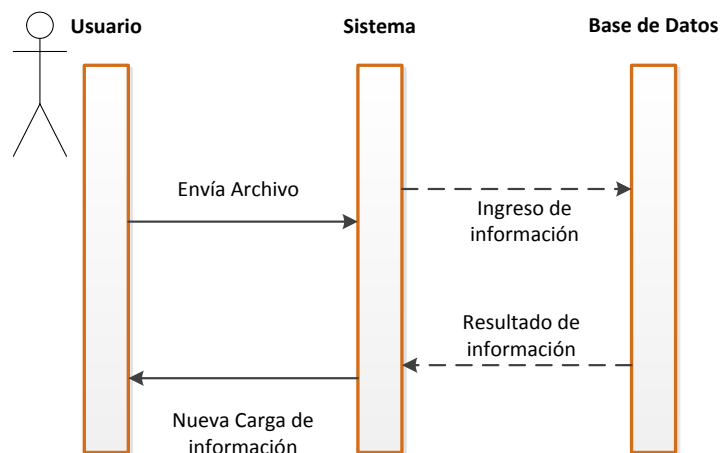
Elaborado por: Livardi Salgado & Daniel Sierra

1.7 Modelo lectura de información

1.7.1 Proceso registro.

Modelo de Caso de Secuencia #15– Lectura

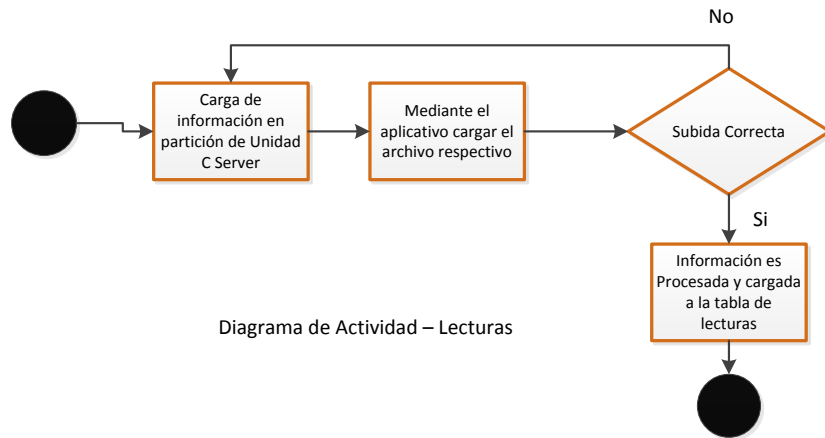
Figura 30. Modelo de Caso de Secuencia, Proceso Lectura



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad # 15 – Lectura

Figura 31. Modelo de Caso de Actividad, Proceso Lectura



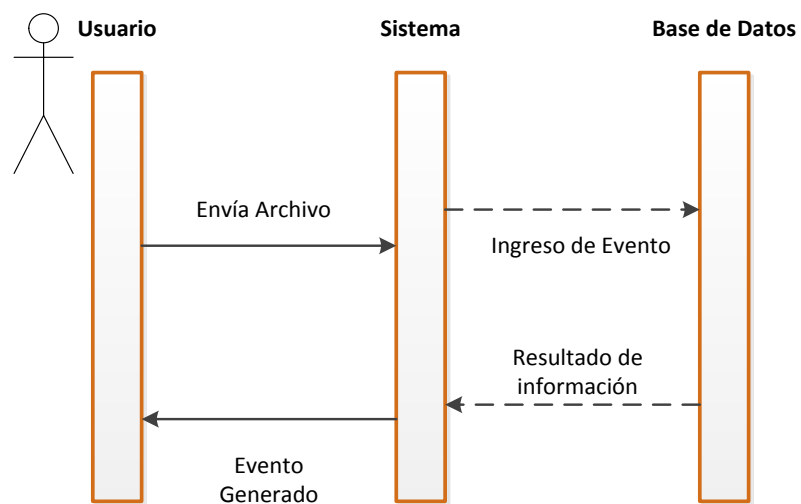
Elaborado por: Livardi Salgado & Daniel Sierra

1.8 Modelo eventos

1.8.1 Proceso creación.

Modelo de Caso de Secuencia #16 – Creación Eventos

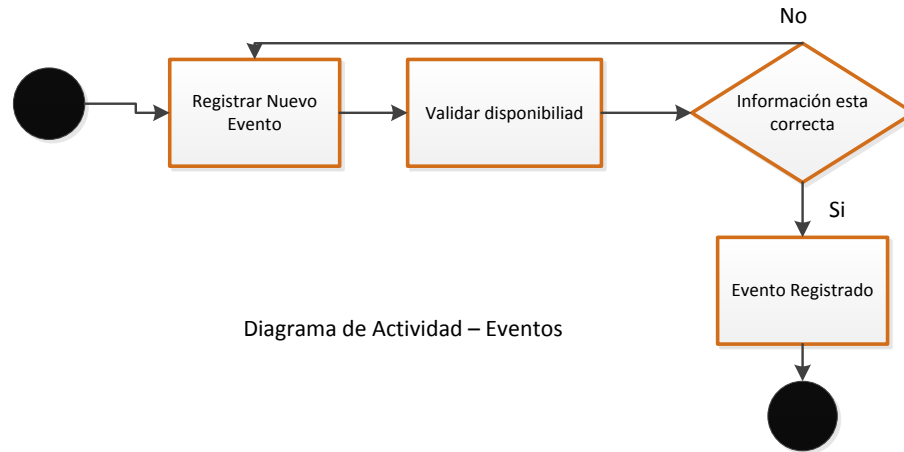
Figura 32. Modelo de Caso de Secuencia, Proceso Creación de Evento



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #16 – Creación Eventos

Figura 33. Modelo de Caso de Actividad, Proceso Creación de Evento



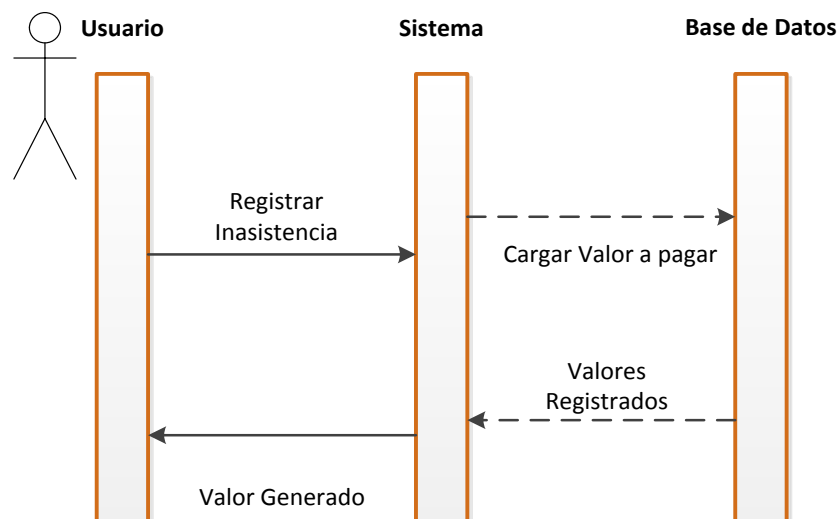
Elaborado por: Livardi Salgado & Daniel Sierra

1.9 Modelo multas

1.9.1 Proceso creacion de multas

Modelo de Caso de Secuencia #17 – Creación de Multas

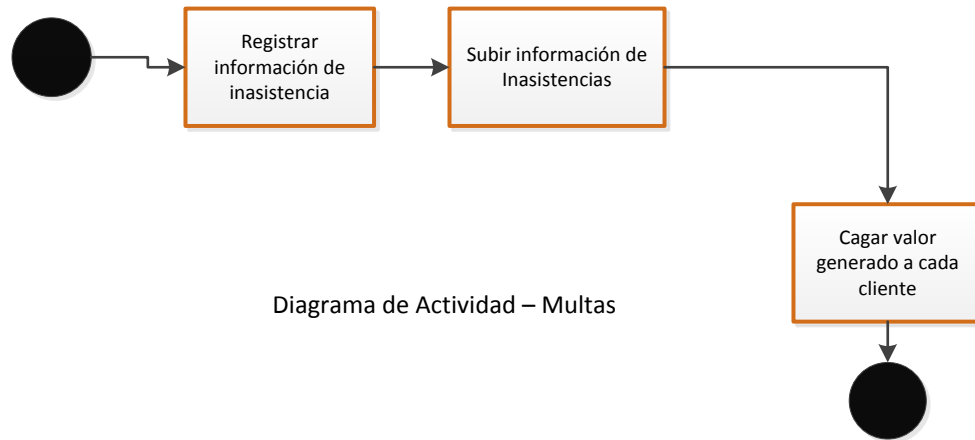
Figura 34. Modelo de Caso de Secuencia, Proceso Creación de Multas



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #17 – Creación de Multas

Figura 35. Modelo de Caso de Actividad, Proceso Creación de Multas



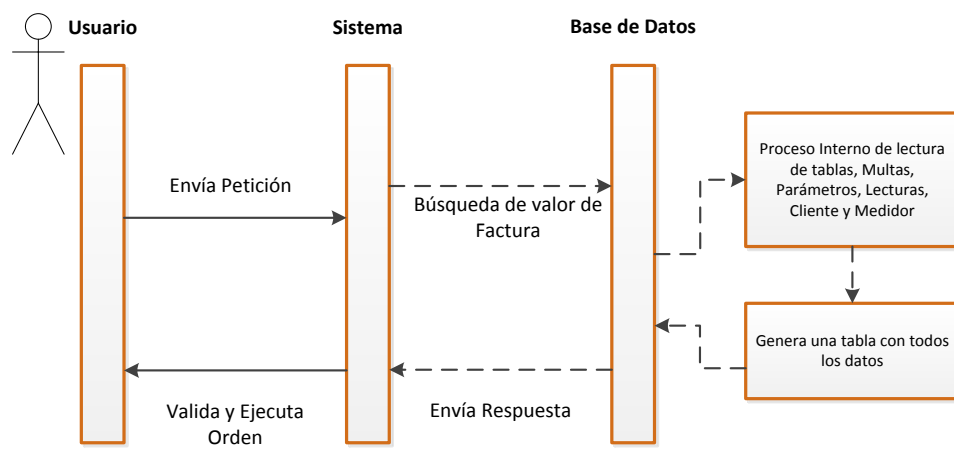
Elaborado por: Livardi Salgado & Daniel Sierra

1.10 Modelo facturas

1.10.1 Proceso creación.

Modelo de Caso de Secuencia #18 – Proceso Facturación

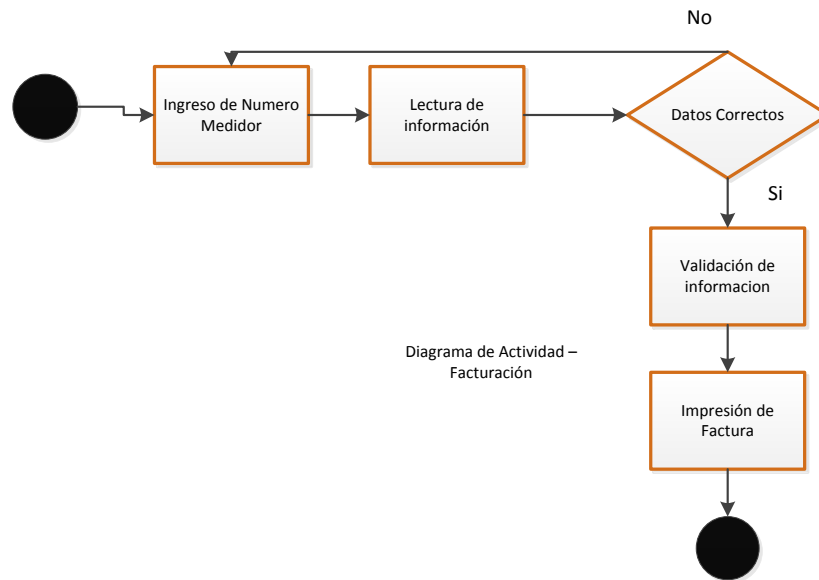
Figura 36. Modelo de Caso de Secuencia, Proceso Facturación



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #18 – Proceso Facturación

Figura 37. Modelo de Caso de Actividad, Proceso Facturación



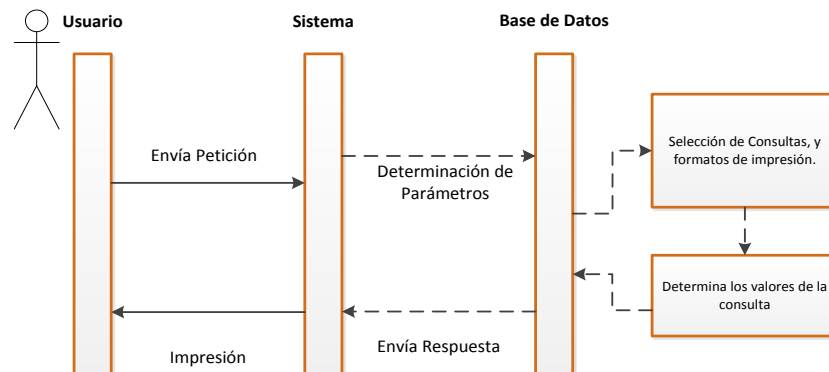
Elaborado por: Livardi Salgado & Daniel Sierra

1.11 Modelo reportes

1.11.1 Proceso generación.

Modelo de Caso de Secuencia # 19 – Generación de Reportes

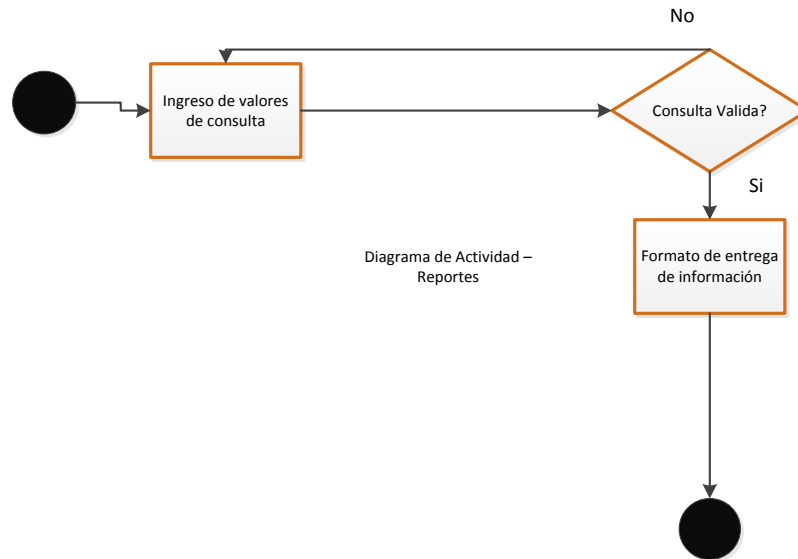
Figura 38. Modelo de Caso de Secuencia – Generación de Reportes



Elaborado por: Livardi Salgado & Daniel Sierra

Modelo de Caso de Actividad #19 – Generación de Reportes

Figura 39. Modelo de Caso de Actividad – Generación de Reportes



Elaborado: Livardi Salgado, Daniel Sierra



SISTEMA DE FACTURACIÓN

PROYECTO PESILLO IMBABURA

Anexo 2. Instalación y configuración de sistemas base

ÍNDICE

INSTALACIÓN Y CONFIGURACIÓN DE SISTEMAS BASE	1
2.1 Firefox.....	1
2.1.1 Introducción.	1
2.1.2 Instalación de Firefox.	1
2.1.3 Búsqueda del instalador.	1
2.1.4 Descarga e instalación.....	1
2.1.5 Ingreso y utilización.	6
2.2 Apache Tomcat	7
2.2.1 Introducción.	7
2.2.2 Objetivo.....	7
2.2.3 Instalación.....	7
2.3 Software de base de datos Postgres.....	12
2.3.1 Introducción.	12
2.3.2 Instalación.....	12
2.3.3 Configuración.....	16

INSTALACIÓN Y CONFIGURACIÓN DE SISTEMAS BASE

2.1 Firefox

2.1.1 Introducción.

Mozilla Firefox es un navegador web, desarrollado para sistemas operativos Windows, Mac OS X y Linux, una de sus ventajas es la búsqueda progresiva, posee un administrador de descargas y una navegación de multi pestañas, que permiten facilidad al usuario para abrir varios contenidos.

2.1.2 Instalación de Firefox.

Con el fin de realizar la instalación del navegador web, se debe realizar los siguientes pasos:

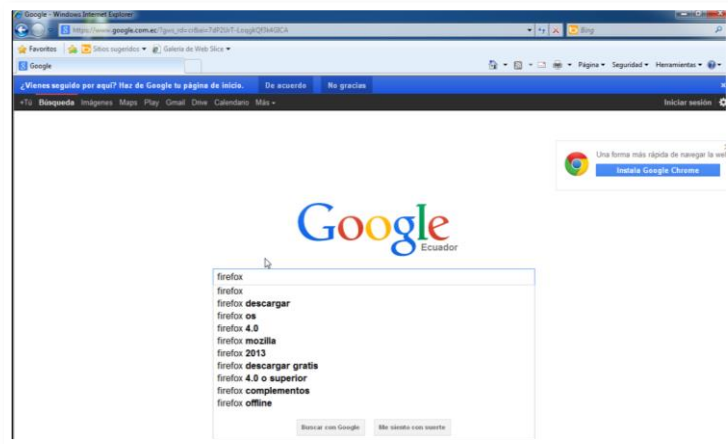
2.1.3 Búsqueda del instalador.

Para iniciar la búsqueda del navegador de Firefox se debe ingresar por otro navegador disponible en nuestro caso es INTERNET EXPLORER.

Una vez ingresados en el menú Inicio >> Internet Explorer.

Se obtiene la siguiente pantalla, en el filtro de google se debe digitar Firefox,

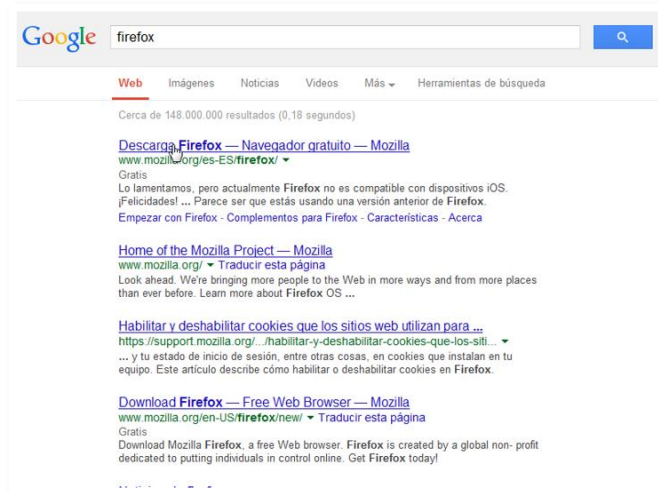
Figura 1. Búsqueda de instalador



Elaborado por: Livardi Salgado & Daniel Sierra

El buscador de google realizara la búsqueda en todos sus motores, y se debe seleccionar la primera opción.

Figura 2. Búsqueda mediante Navegador



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez que se ha seleccionado la primera opción el navegador redirige a la siguiente pantalla.

Dentro de esta página web se encuentra el instalador de Firefox,

Figura 3. Descarga del instalador WEB



Elaborado por: Livardi Salgado & Daniel Sierra

2.1.4 Descarga e instalación.

Una vez ingresado al instalador se obtiene el siguiente mensaje se debe seleccionar ejecutar.

Figura 4. Descarga del ejecutable



Elaborado por: Livardi Salgado & Daniel Sierra

Después se obtiene un nuevo mensaje de seguridad de Windows, se debe seleccionar Ejecutar

Figura 5. Ejecutable de Firefox



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez concluido se obtiene el siguiente mensaje, es la pantalla de inicio al instalador de Firefox, se debe realizar un clic sobre Instalar

Figura 6. Instalación Firefox



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez realizado la actividad anterior el instalador empieza a descargar y a registrar

sobre el sistema la aplicación.

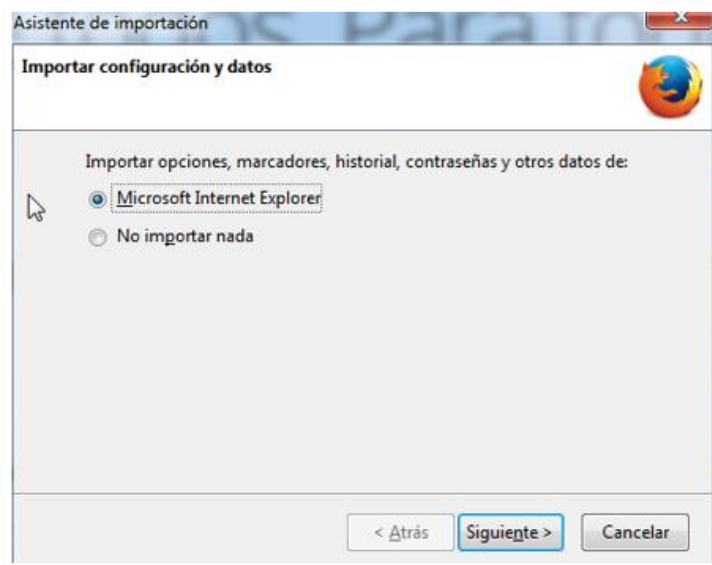
Figura 7. Descarga e Instalación



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez terminada la pantalla anterior, el mensaje mencionara si “desea importar las direcciones utilizadas últimamente de los otros navegadores”

Figura 8. Importación de historial

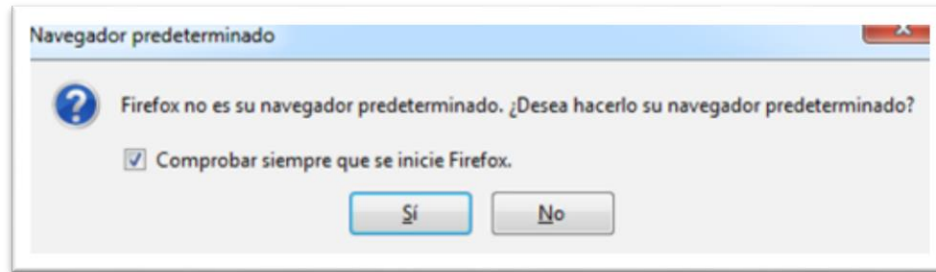


Elaborado por: Livardi Salgado & Daniel Sierra

2.1.5 Ingreso y utilización.

Una vez finalizada, el sistema indicara si desea registrarlo como predeterminado.

Figura 9. Predeterminar el navegador



Elaborado por: Livardi Salgado & Daniel Sierra

Finalmente se obtiene le mensaje de bienvenida a Firefox con el cual ya puede utilizarlo con fines de navegación de búsqueda y de acceso al sistema de facturación del Proyecto Pesillo - Imbabura.

Figura 10. Funcionamiento Firefox



Elaborado por: Livardi Salgado & Daniel Sierra

2.2 Apache Tomcat

2.2.1 Introducción.

El servicio Apache Tomcat establece la comunicación entre el sistema y la base datos que permite la publicación web del sistema mediante un navegador Web.

2.2.2 Objetivo.

Instalar y configurar el servicio web para el levantamiento y uso del sistema mediante un sitio web en un navegador Open Source

2.2.3 Instalación.

El instalador de apache será colocado en los instaladores del proyecto.

Figura 11. Instalador base



Elaborado por: Livardi Salgado & Daniel Sierra

Al realizar doble clic sobre el acceso del instalador, se obtiene la siguiente pantalla que menciona el inicio de la instalación.

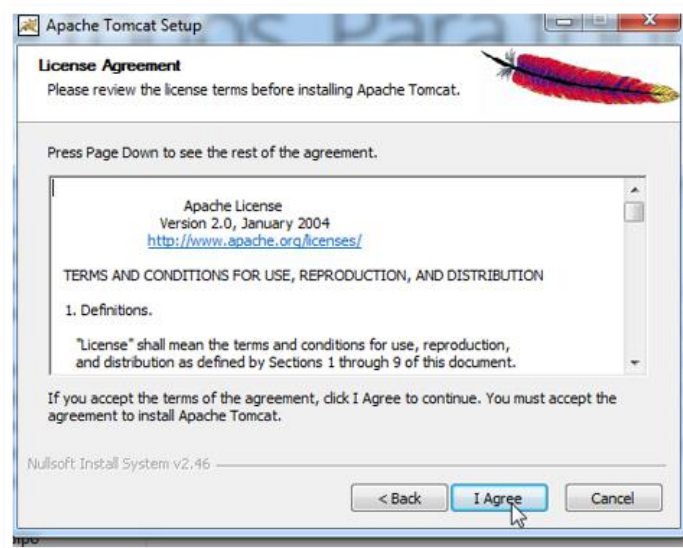
Figura 12. Pantalla inicial de la instalación



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez realizado el clic sobre siguiente, se obtiene la pantalla de licencia se debe realizar un click sobre **I AGREE**

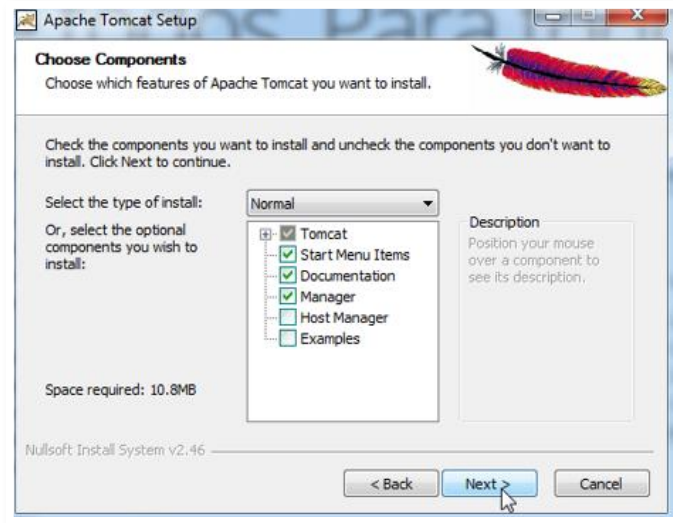
Figura 13. Contrato de seguridad de Apache



Elaborado por: Livardi Salgado & Daniel Sierra

A continuación se recibe la pantalla de componentes del apache, la misma que se debe realizar clic sobre Siguiente.

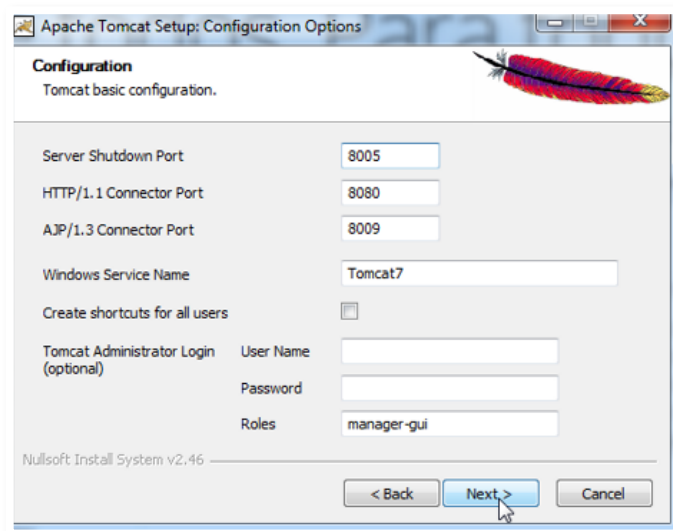
Figura 14. Propiedades y características



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez seleccionado el siguiente se debe verificar los puertos que se abren con el servicio de apache en nuestro caso es el 8082

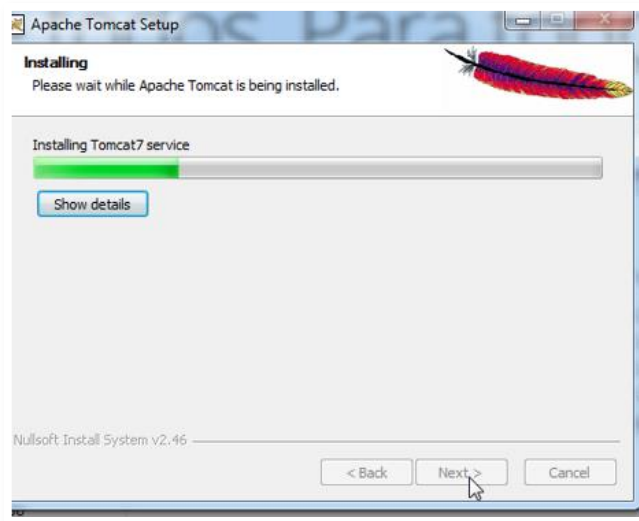
Figura 15. Configuración del puerto



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez realizado el paso anterior el instalador empieza a registrar la información necesaria.

Figura 16. Proceso de instalación



Elaborado por: Livardi Salgado & Daniel Sierra

El proceso finaliza al momento de obtener la información de instalación completa en el cual se debe seleccionar el cuadro “RUN APACHE TOMCAT” con el fin de iniciar el servicio web.

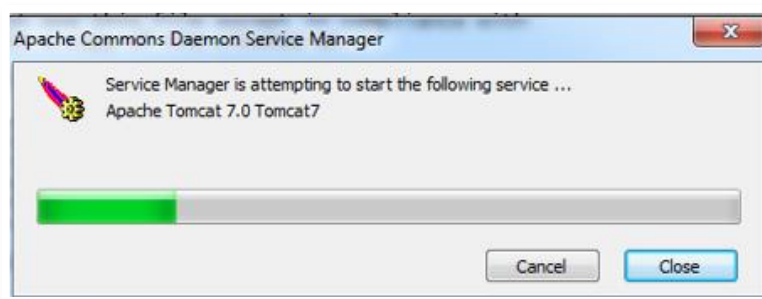
Figura 17. Instalación finalizada



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez realizado el clic sobre siguiente se obtiene la siguiente figura que menciona el inicio y estado del sistema.

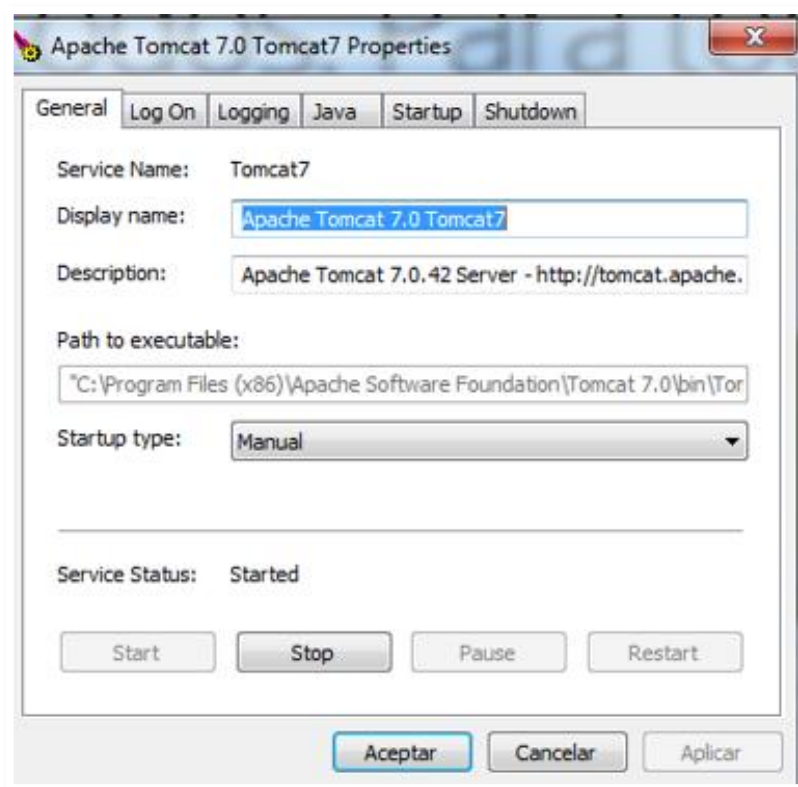
Figura 18. Levantamiento del servicio



Elaborado por: Livardi Salgado & Daniel Sierra

Finalmente muestra el estado del servicio al finalizar la instalación.

Figura 19. Estado del servicio



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez finalizada la instalación de Apache continuamos con la instalación y configuración de la base de datos Postgres.

2.3 Software de base de datos Postgres

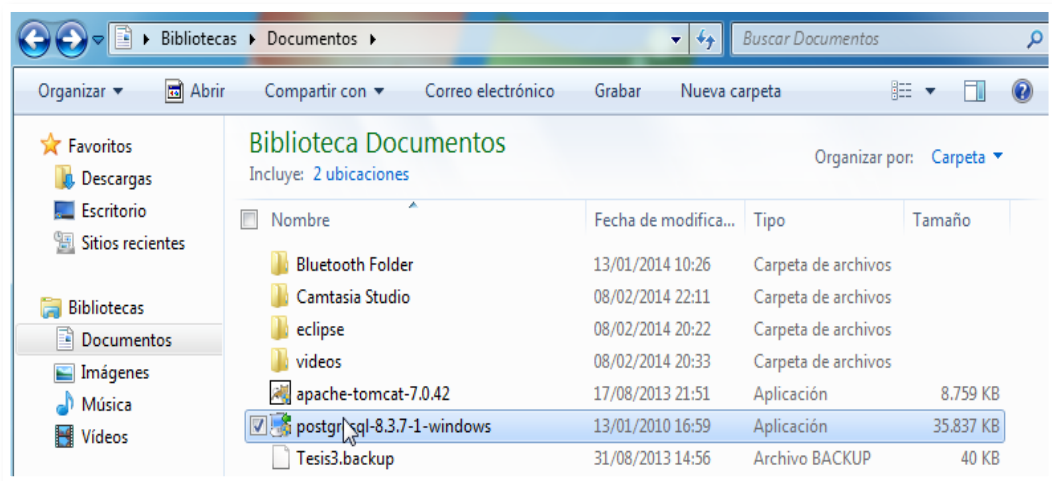
2.3.1 Introducción.

La instalación y configuración de la base de datos permite que el sistema reciba y guarde los datos procesados, estos datos son indispensables para el proyecto.

2.3.2 Instalación.

Es necesario localizar el instalador de la base de datos, denominado Postgresql:

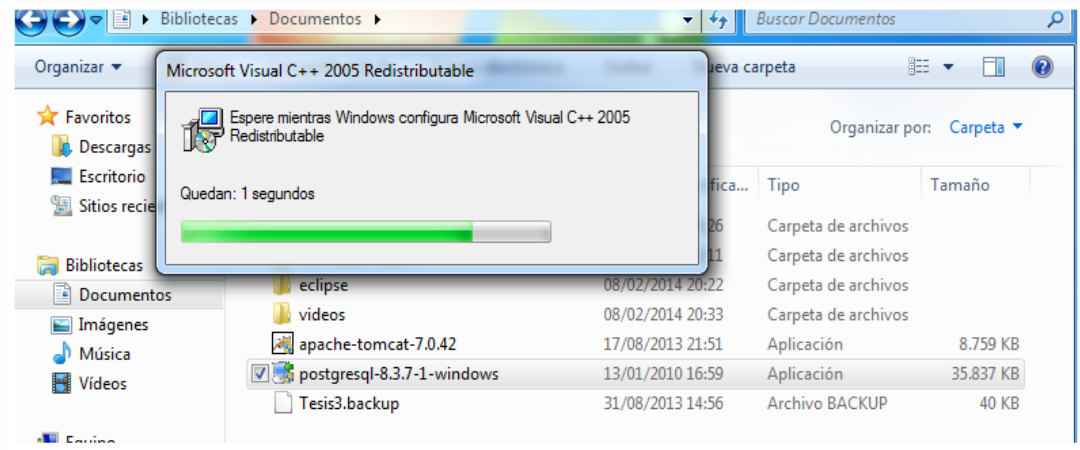
Figura 20. Instalación de Postgres



Elaborado por: Livardi Salgado & Daniel Sierra

Se ejecuta la instalación del archivo seleccionado previamente:

Figura 21. Proceso de instalación



Elaborado por: Livardi Salgado & Daniel Sierra

Pantalla inicial del proceso de instalación de la base de datos.

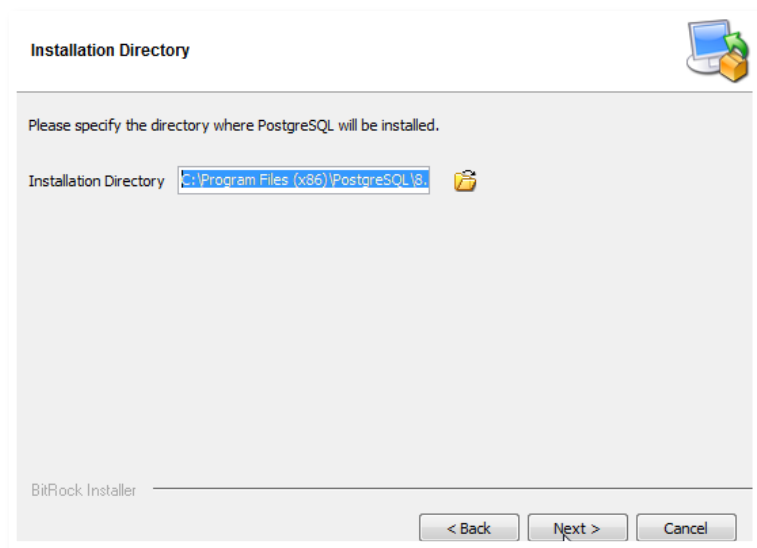
Figura 22. Inicio de la instalación



Elaborado por: Livardi Salgado & Daniel Sierra

A continuación, se debe seleccionar el “path” o directorio de la instalación:

Figura 23. Directorio de instalación

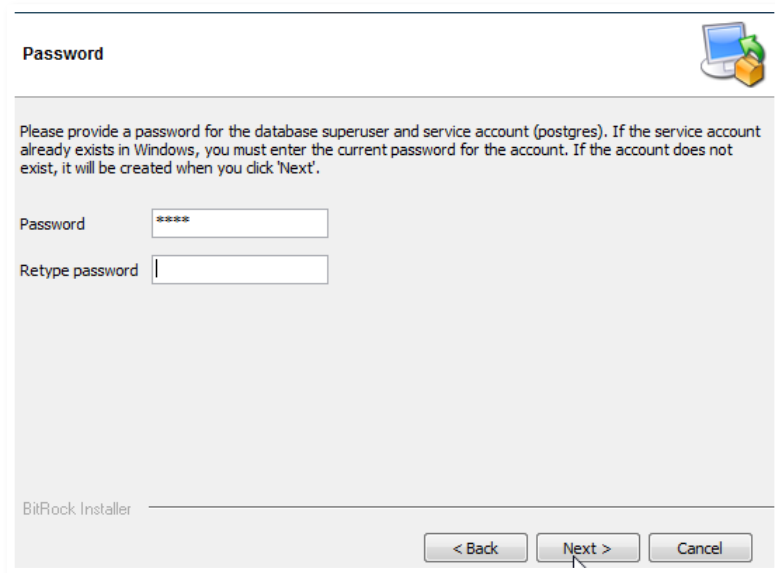


Elaborado por: Livardi Salgado & Daniel Sierra

El sistema solicitará la creación de una clave de acceso a la plataforma:

La clave actual es: ppiups2014

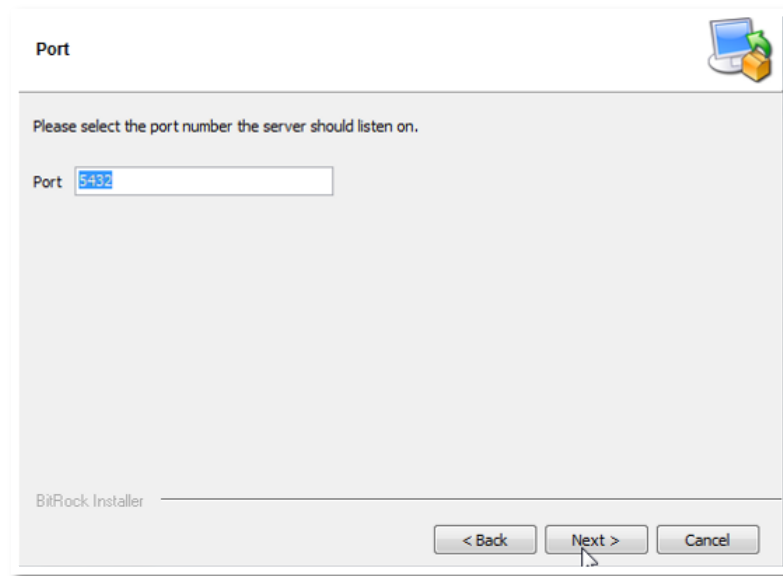
Figura 24. Clave de base de datos



Elaborado por: Livardi Salgado & Daniel Sierra

El puerto, es la vía por la cual se establece la comunicación de la base de datos:

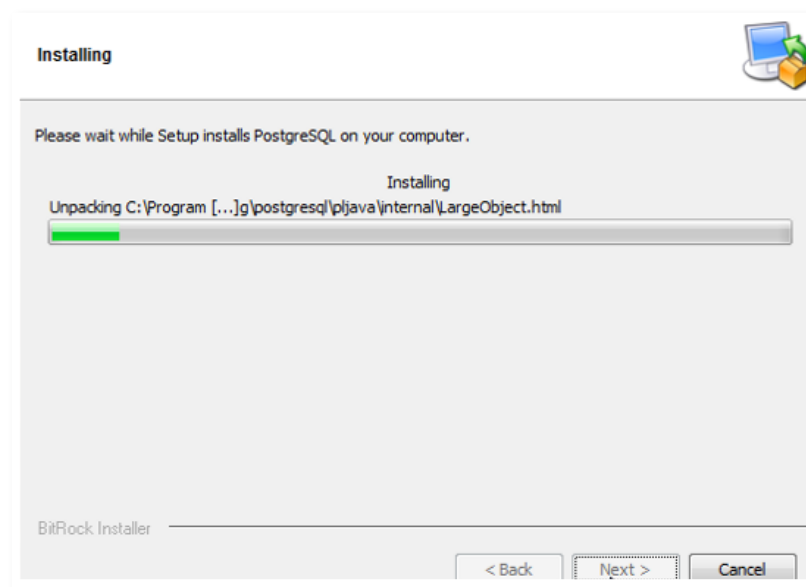
Figura 25. Selección del Puerto



Elaborado por: Livardi Salgado & Daniel Sierra

El sistema mostrara el avance de la instalación:

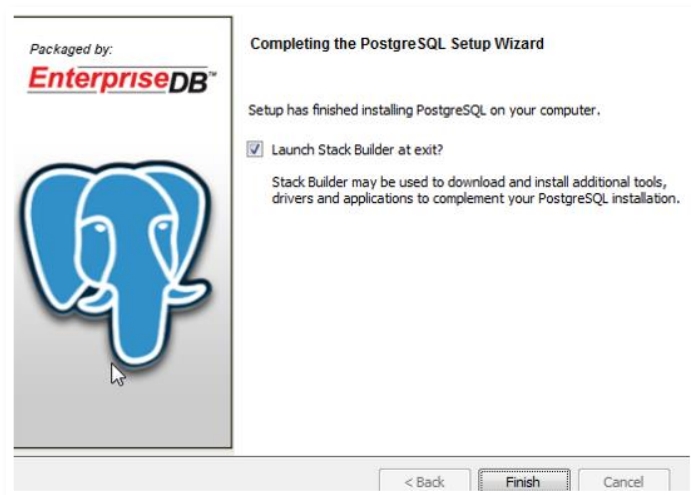
Figura 26. Proceso de instalación



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez finalizada la instalación, se debe dar por concluido.

Figura 27. Finalización de instalación

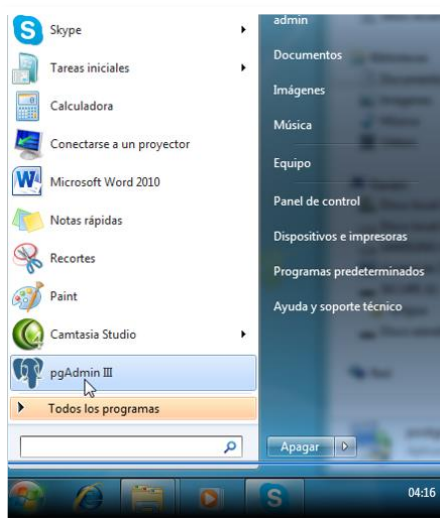


Elaborado por: Livardi Salgado & Daniel Sierra

Al finalizar la instalación de la base de datos, se creará un acceso a la base de datos se denominara "PgAdmin III"

2.3.3 Configuración.

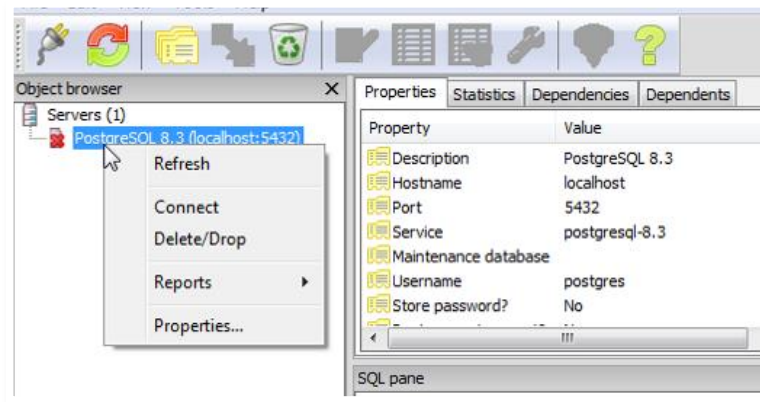
Figura 28. Acceso directo a PostgreSQL



Elaborado por: Livardi Salgado & Daniel Sierra

Al ingresar a PostgreSQL, se encuentra la opción de conectar con la base de datos:

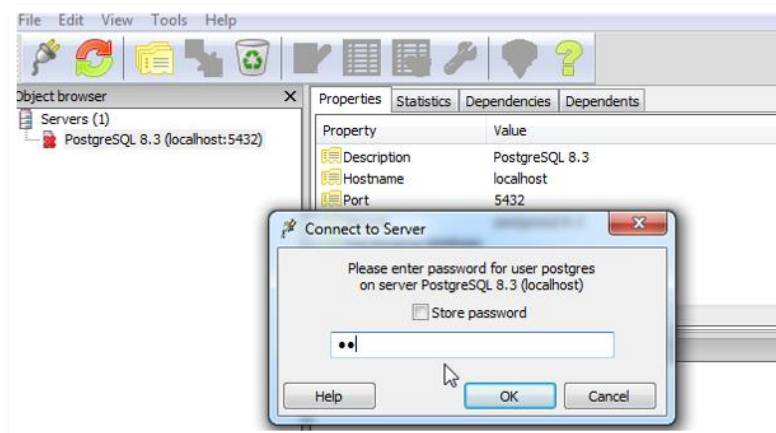
Figura 29. Conexión a la Base



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresada la clave, el sistema permitirá la creación de una base de datos, como se muestra en la imagen a continuación:

Figura 30. Validación de clave BDD



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresada la clave, el sistema permitirá la creación de una base de datos,

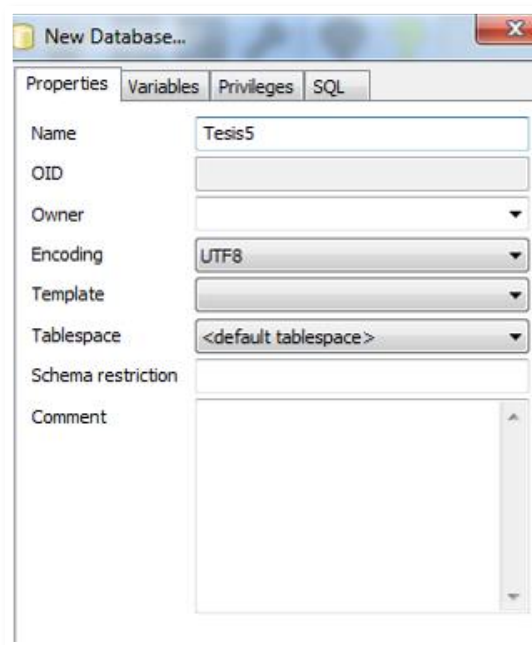
Figura 31. Creación de BDD



Elaborado por: Livardi Salgado & Daniel Sierra

Finalmente, se debe crear la base de datos para relacionarlo con el sistema:

Figura 32. Creación y registro de BDD



Elaborado por: Livardi Salgado & Daniel Sierra

Anexo 3. Manual de usuario

ÍNDICE

MANUAL DE USUARIO.....	1
3.1 Introducción.....	1
3.2 Inicio	1
3.3 Interfaz de usuario	3
3.4 Navegación interfaz.....	3
3.4.1 Login.....	3
3.4.2 Perfiles de navegación.....	4
3.4.3 Módulo usuario.	5
3.4.4 Módulo administración de usuario.....	5
3.4.5 Módulo de clientes.....	6
3.4.6 Módulo creación de perfiles.	8
3.4.7 Módulo comunidades.....	9
3.4.8 Módulo sectores.....	10
3.4.9 Módulo medidor.....	11
3.4.10 Módulo de carga de registros.....	12
3.4.11 Módulo de parámetros.....	13
3.4.12 Módulo de facturación.....	14
3.4.13 Módulo de reportes.	16

MANUAL DE USUARIO

3.1 Introducción

El presente documento describe de manera clara y concisa cómo se utiliza el sistema de facturación proyecto Pesillo Imbabura para el control, registro y facturación del uso de agua.

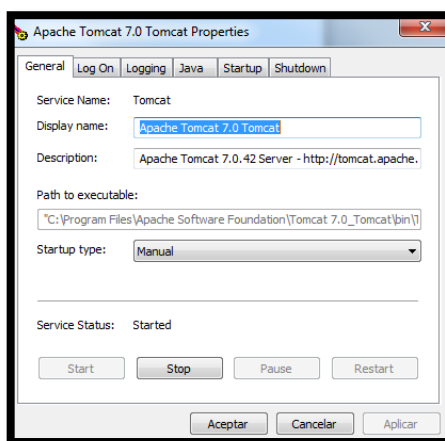
El sistema fue desarrollado con la finalidad de controlar el uso y pago del consumo de agua de las comunidades del proyecto.

Es importante consultar este manual antes y durante del uso del sistema, para conocer las funcionalidades y proceso de facturación.

3.2 Inicio

Luego de encender el computador que contiene el sistema de facturación configurado; se debe verificar que el proceso principal denominado “Servicio Tomcat” se encuentre activo, el sistema hace la funcionalidad de “Cliente – Servidor” la forma de verificación se muestra en el figura:

Figura 1. Servicio Tomcat



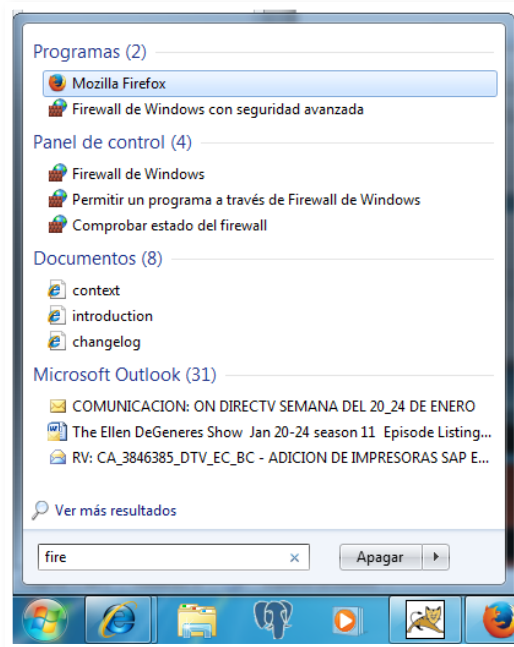
Elaborado por: Livardi Salgado & Daniel Sierra

Una vez revisado que el estado del servicio este “STARTED – INICIADO” se procede a abrir el navegador Web.

Navegación web

Para el uso del sistema se debe acceder a través del navegador Firefox, como se demuestra mediante la figura a continuación:

Figura 2. Pantalla Servicio Mozilla

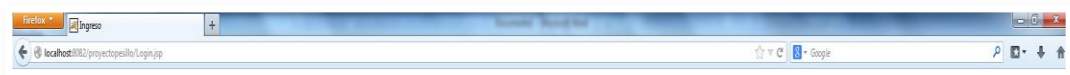


Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresado al navegador Firefox de debe digitar la siguiente dirección como se indica en la figura 3:

<http://localhost:8082/proyectopesillo/Login.jsp>

Figura 3. Dirección web de acceso al sistema



Elaborado por: Livardi Salgado & Daniel Sierra

3.3 Interfaz de usuario

Son las ventanas, mensajes y opciones que el usuario va a manejar durante el uso del sistema, comprende los puntos de contacto entre el usuario y el equipo.

3.4 Navegación interfaz

3.4.1 Login.

Para ingresar al sistema es necesario ingresar la identificación del usuario:

Figura 4. Pantalla de Inicio

The image shows a web application interface. At the top is a banner image of a mountain landscape with a lake and a bright sun. Below the banner is a blue header bar with the text 'SISTEMA DE FACTURACION PESILLO IMBABURA'. Underneath is a red section titled 'IDENTIFICACION DE USUARIO'. This section contains two input fields: 'User:' with a user icon and 'Clave:' with a key icon. To the right of the 'Clave:' field is a green checkmark icon. Below these fields is a button labeled 'enviar'. At the bottom of the red section is a dark grey bar with the text 'OTRAS INFORMACIONES' and a link labeled 'Portada'.

Elaborado por: Livardi Salgado & Daniel Sierra

La ventana de login consta de dos campos de texto:

Usuario: El campo usuario es la identificación de la persona para acceder al sistema.

Clave: La clave es una serie secreta de caracteres que permite al usuario tener acceso al sistema.

Una vez ingresados los campos Usuario y Clave el sistema realizará unos de los dos posibles procesos:

El acceso correcto al sistema, posterior a la validación de los campos, si los mismos son correctos. Se muestra en la figura.

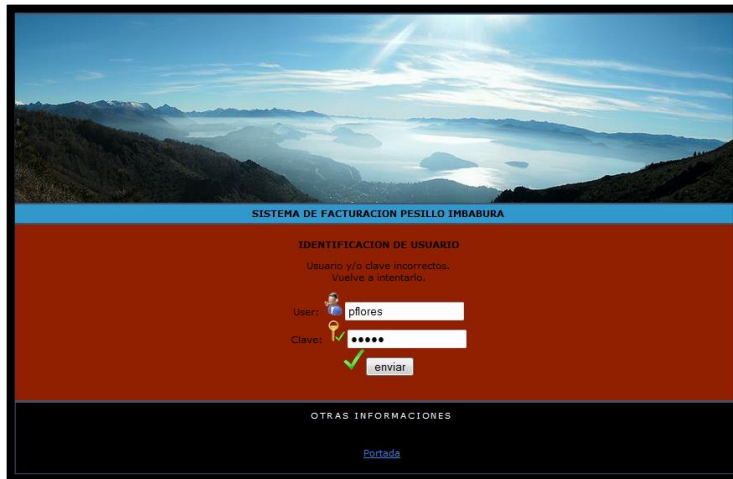
Figura 5. Acceso correcto al sistema



Elaborado por: Livardi Salgado & Daniel Sierra

Si alguno de los dos campos ingresados es incorrecto, no es posible la validación de los mismos como se presenta el siguiente mensaje:

Figura 6. Acceso incorrecto



Elaborado por: Livardi Salgado & Daniel Sierra

3.4.2 Perfiles de navegación.

Administrador: Este tipo de usuarios tienen acceso total al sistema y a todas sus opciones:

- Usuarios
- Clientes
- Medidor
- Parámetros
- Multas
- Reportes
- Facturación

3.4.3 Módulo usuario.

El usuario podrá adquirir perfiles previamente asignados por un administrador, e incluso un usuario puede llegar a ser un administrador con permisos previamente concedidos.

3.4.4 Módulo administración de usuario.

La administración de usuarios es posible solo a través de un usuario administrador, el cual tiene las funciones de crear, modificar y eliminar los usuarios que pueden acceder al sistema.

Creación de usuario.

Una vez ingresado en el sistema se visualizarán varias opciones para registrar diferentes tipos de información:

- Usuarios
- Clientes
- Medidores
- Parámetros
- Multas
- Facturación
- Reportes

3.4.5 Módulo de clientes.

Dentro del módulo de clientes existen las opciones de Creación, Modificación y Eliminación de usuarios.

Figura 7. Menú clientes



Elaborado por: Livardi Salgado & Daniel Sierra

Ingreso de cliente

El ingreso es una secuencia de pasos necesarios para el registro, el nuevo cliente debe ser asignado a una comunidad del sector como se muestra en la figura:

Figura 8. Ingreso de comunidad



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez seleccionada la comunidad, el sistema re direcciona el registro para el ingreso de datos del cliente, se muestra en la figura:

Figura 9. Registro de cliente

Cedula: 1312321498
Sector: Santa Clara
Nombre: Ma Alejandra
Direccion: Orozco
Tel. Domicilio: 023456789
Tel. Celular: 0932156489
Registrar

INGRESE LA INFORMACION SOLICITADA
[Volver](#)

BIENVENIDO ADMINISTRADOR: PFLORES

SISTEMA DE FACTURACION PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA

Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresados los datos del cliente, se debe dar un click sobre la opción REGISTRAR para guardar la información en la base de datos del sistema. Como confirmación de la información guardada aparecerá el siguiente mensaje “SE PROCESO LA INFORMACIÓN CORRECTAMENTE”, tal se muestra en la siguiente figura:

Figura 10. Ingreso de información

NOTIFICACION

SE PROCESO LA INFORMACION CORRECTAMENTE

[Volver](#)

BIENVENIDO ADMINISTRADOR: PFLORES

SISTEMA DE FACTURACION PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.6 Módulo creación de perfiles.

La opción de creación de perfiles se encuentra asignada al administrador del sistema, quien faculta el acceso a los diferentes módulos a los usuarios registrados.

La ventana de Perfiles contiene la opción Nuevo y Eliminar.

Figura 11. Creación de perfiles



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresado a la opción Nuevo, se obtendrá la siguiente pantalla,

Figura 12. Asignación de perfiles



Elaborado por: Livardi Salgado & Daniel Sierra

3.4.7 Módulo comunidades.

El modulo comunidades contiene la opción de creación, modificación y eliminación de comunidades existentes.

Para la creación de una comunidad, es necesario el ingreso de los siguientes datos:

Nombre Comunidad: Nombre que se otorgara a esa junta

Descripción: Breve detalle de la comunidad

Representante: Jefe político de la comunidad

Ubicación: Detalle de la zona.

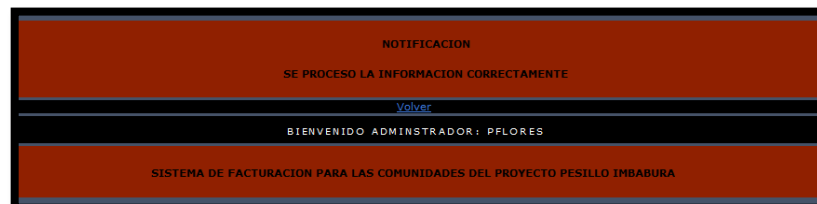
Figura 13. Registro de Comunidades



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez guardada la información aparecerá la imagen

Figura 14. Información guardada



Elaborado por: Livardi Salgado & Daniel Sierra

3.4.8 Módulo sectores.

El modulo sector será usado para subdividir la comunidad con el fin de optimizar el proceso de registro de la información.

Para la creación los datos que se solicitan son:

Comunidad: Se debe seleccionar una comunidad existente.

Nombre: Nombre del nuevo sector

Habitantes: Se debe señalar el número de habitantes.

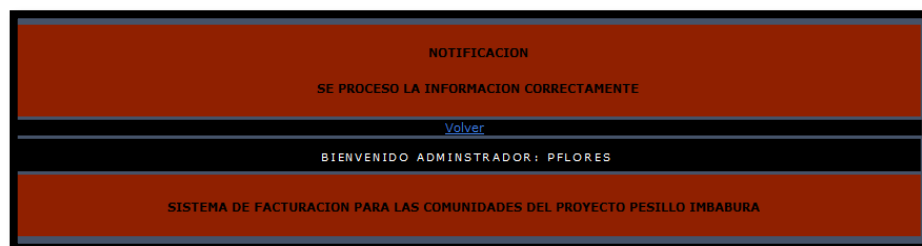
Figura 15. Registro de sectores

Figura 15. Registro de sectores

Elaborado por: Livardi Salgado & Daniel Sierra

Una vez guardada la información aparecerá la imagen

Figura 16. Registro ingresado



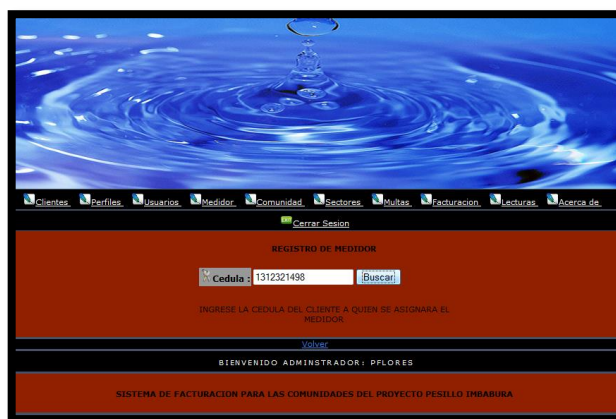
Elaborado por: Livardi Salgado & Daniel Sierra

3.4.9 Módulo medidor.

Para el registro de un medidor de agua, es necesario que consten previamente los datos en todos los módulos con el fin de agilizar la carga de información.

En el módulo medidor se solicita la digitación del nuevo número de cédula de identidad para el registro.

Figura 17. Registro de medidor



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresado el número de cédula, el sistema avanzará a la ventana de datos, este módulo proporciona los datos extraídos desde la base de datos analizados por el número de cédula

Figura 18. Registro de medidor por cliente

REGISTRO DE MEDIDOR POR CLIENTE

Cedula :	1312321488
Nombre :	Mu Alejandra
Direccion :	Orocco
Telefono :	02468789
Comunidad :	Cayambe
Sector :	Santa Clara
Servicio :	Residencial
Modelo :	0008
Tiempo de Vida :	6
Costador Inicial :	00000003
Fecha de Ingreso :	2014-01-21

[Registrar](#)

Elaborado por: Livardi Salgado & Daniel Sierra

Una vez guardada la información aparecerá la imagen

Figura 19. Registro de información

NOTIFICACION

SE PROCESO LA INFORMACION CORRECTAMENTE

[Volver](#)

BIENVENIDO ADMINISTRADOR: PFLORES

SISTEMA DE FACTURACION PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.10 Módulo de carga de registros.

Este módulo se encarga de subir automáticamente el directorio que se encuentra almacenado en el equipo en un formato especial, la información de la medición de uso de cada medidor, tiene el fin de registrar la información entregada por el personal de registro.

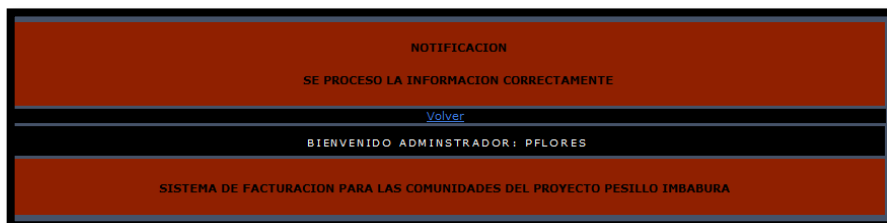
Figura 20. Carga de lecturas



Elaborado por: Livardi Salgado & Daniel Sierra

Una vez guardada la información aparecerá la imagen.

Figura 21. Ingreso satisfactorio



Elaborado por: Livardi Salgado & Daniel Sierra

3.4.11 Módulo de parámetros.

Este módulo se encarga directamente del control de valores por consumo de m³ de agua de cada comunidad con el fin de proporcionar al administrador la opción de crear sus valores.

Figura 22. Registro de parámetros

REGISTRO DE PARAMETROS	
Comunidad	Valle Alto
Reg. Minimo	
Reg. Maximo	
Valor del m3	
Valor de instalacion nuevo medidor	
Valor de la multa	
Meses a diferir	
<input type="button" value="Registrar"/>	

[Volver](#) [Portada](#)
© BY LIVARDI

Elaborado por: Livardi Salgado & Daniel Sierra

Una vez guardada la información aparecerá la imagen.

Figura 23. Información Guardada

NOTIFICACION
SE PROCESO LA INFORMACION CORRECTAMENTE

[Volver](#)

BIENVENIDO ADMINSTRADOR: PFLORES

SISTEMA DE FACTURACION PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA

Elaborado por: Livardi Salgado & Daniel Sierra

3.4.12 Módulo de facturación.

Permite realizar el cobro en base al procedimiento realizado anteriormente, este módulo está asociado a todos los módulos del sistema con el fin que la información sea confiable y concreta.

Figura 24: Modulo de facturación

Elaborado por: Livardi Salgado & Daniel Sierra

Una vez ingresada la información solicitada por el módulo, se envía a generar el cobro con el cual se obtiene el siguiente cuadro:

Figura 25: Impresión de factura

PROYECTO PESILLO IMBABURA	
REPORTE DE FACTURAS PENDIENTES	
FACTURA	
DATOS DEL CLIENTE	DETALLE
Factura: FACT-0	Consumo: 0
Suministro: 0	Alcantarillado: 0
Servicio: 0	Int. x Diferido: 0
Cédula: 0	Int x Mora: 0
Nombre: 0	Otros: 0
Dirección: 0	Subtotal: 0
	I.V.A.: 0
	Saldo Pendiente: 0
	Total: 0
	Valor a Pagar: 0
Estado: PENDIENTE	F. de Emisión: 3/14/14 8:23 PM

Elaborado por: Livardi Salgado & Daniel Sierra

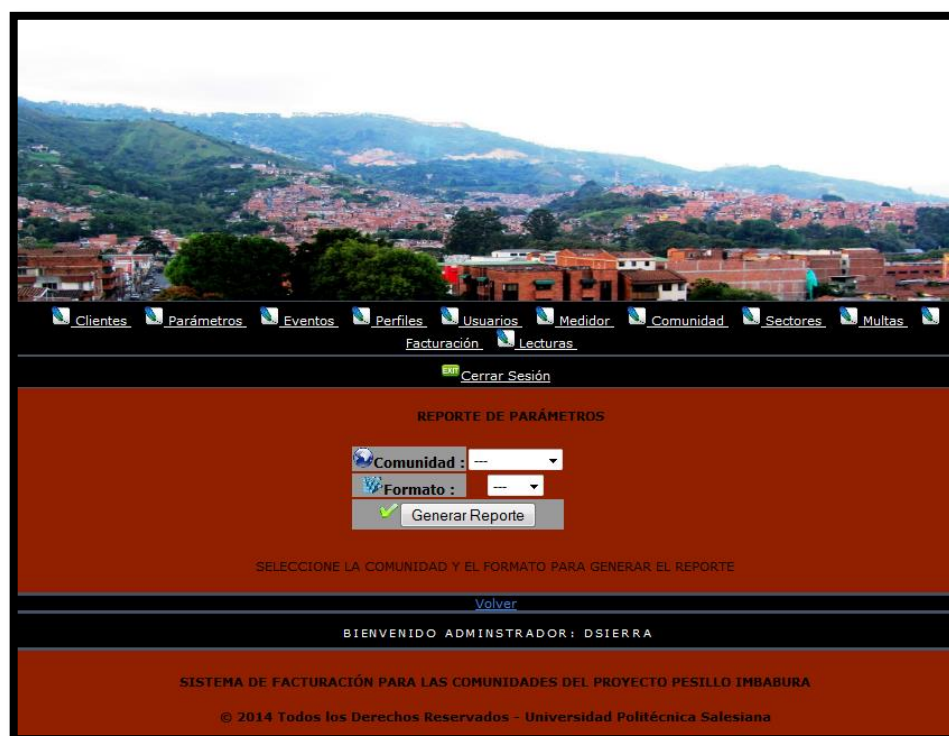
Esta es la planilla final del sistema de facturación con el cual se establece el cobro al abonado

Este manual de usuario permite tener una idea más clara del sistema mencionado

3.4.13 Módulo de reportes.

Permite obtener la información de clientes, medidor, usuarios, administradores y facturación.

Figura 26: Modulo de reportes



Elaborado por: Livardi Salgado & Daniel Sierra

Todas las funciones de los sistemas han sido análisis de acuerdo a las necesidades presentadas mediante la Junta de acción del proyecto Pesillo Imbabura, con el fin de realizar sus operaciones de una manera controlada y sistemática mediante el sistema presentado.

Se agradece la colaboración de todas las personas que formaron parte del Proyecto Pesillo - Imbabura

Figura 27: Proyecto Pesillo - Imbabura



Elaborado por: Livardi Salgado & Daniel Sierra

